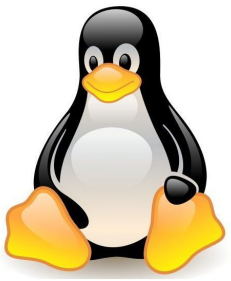
Linux Plus
for
AWS and DevOps



CLARUSWAY
WAY TO REINVENT YOURSELF

Using Package Managers

CLARUSWAY
WAY TO REINVENT YOURSELF

Table of Contents

1

Installing New Software

Installing New Software

Most Linux distributions contain **ample support for video and network cards, monitors and other external devices**, so there is usually **no need to install extra drivers**.

If you just can't find what you need, **maybe it is not installed** on your system. Linux moves fast, and software improves on a daily basis.

The **website of software vendor** is a good place to start looking for instructions about how to install it on your type of Linux.

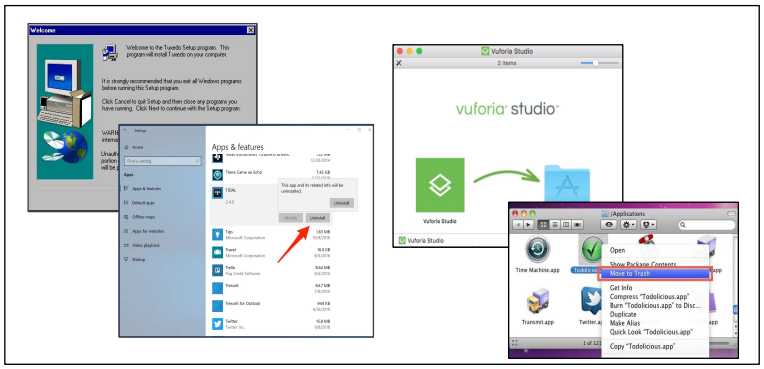
https://git-scm.com/downloads

2 Package Management

Package Management

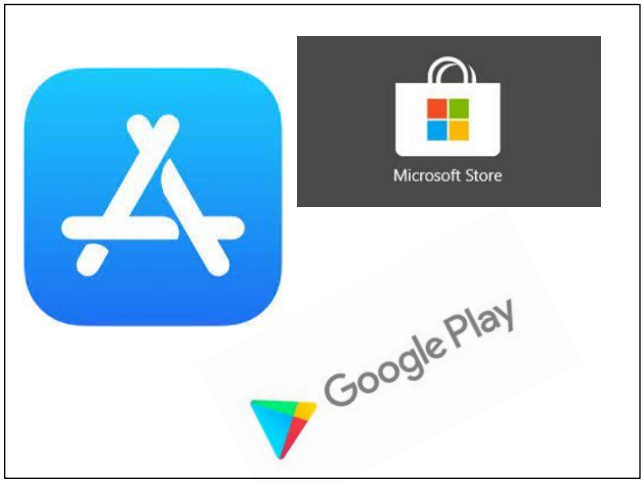
A package manager is a **collection of software tools** that **automates** the process of **installing**, **upgrading**, **configuring**, and **removing software** on a host in a consistent manner.

Windows / Mac Software Installations



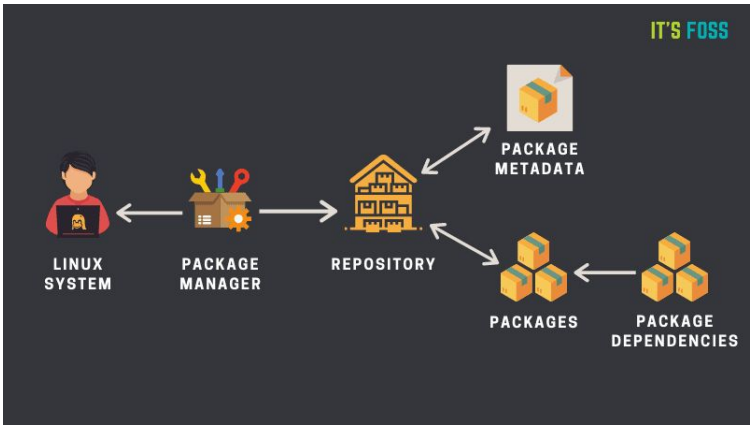
- Windows and MacOS make light work of installing and uninstall software
- The built-in tools manage software installation along with all dependencies

Centralized “Stores” (Repositories)



- Apple’s App Store made it popular to find software in a central location
- Others followed suit, including Microsoft and Google
- These app stores also enable users to easily install and uninstall apps

Package Management



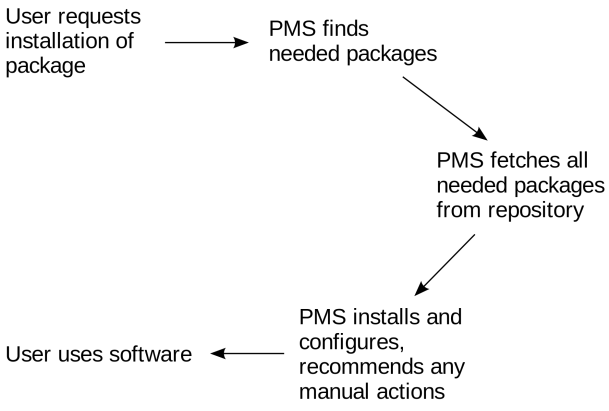
- Package managers have access to centralized repositories (“repos”)
- In addition, they manage installation, removal and software upgrades

Package Management

Package managers typically **maintain a database of software dependencies** and **version information** to **prevent** software **mismatches** and **missing prerequisites**. They work closely with software repositories, binary repository managers, and app stores.

Package Management

Package managers are **designed to eliminate the need for manual installs** and **updates**. This can be particularly useful for large enterprises whose operating systems are typically consisting of **hundreds or even tens** of thousands of distinct software packages.



Package Management

Typical functions of a package manager include:

- **Working with file archivers** to extract package archives
- **Ensuring the integrity** and **authenticity** of the package by verifying their checksums and digital certificates, respectively
- **Looking up**, **downloading**, **installing**, or **updating** existing software from a software repository or app store
- **Managing dependencies** to ensure a package is installed **with all packages it requires**, thus avoiding "**dependency hell**"

Package Management

Operating System	Format	Tool(s)
Debian	`.deb`	`apt`, `apt-cache`, `apt-get`, `dpkg`
Ubuntu	`.deb`	`apt`, `apt-cache`, `apt-get`, `dpkg`
CentOS	`.rpm`	`yum`
Fedora	`.rpm`	`dnf`
FreeBSD	Ports, `.txz`	`make`, `pkg`

https://stackoverflow.com/questions/10286459/multiple-package-manager

3 Package Terminology

CLARUSWAY
WAY TO REINVENT YOURSELF

Package Terminology

- Repository : A lot of software and documentation for your Linux distribution is available as packages in one or more centrally distributed repositories.

"A few years ago, before the proliferation of smartphones, the idea of a software repository was difficult for many users to grasp if they were not involved in the Linux ecosystem. To this day, most Windows users still seem to be hardwired to open a web browser to search for and install new software. However, those with smartphones have gotten used to the idea of a software "store." The way smartphone users obtain software and the way package managers work are not dissimilar. While there have been several attempts at making an attractive UI for software repositories, the vast majority of Linux users still use the command line to install packages. Software repositories are a centralized listing of all of the available software for any repository the system has been configured to use."

(https://opensource.com/article/18/7/evolution-package-managers)

CLARUSWAY
WAY TO REINVENT YOURSELF

Package Terminology

- Repository

```
user@arch ~ $  aurman -Ss kate
```

```
[user@centos ~]$ yum search kate
```

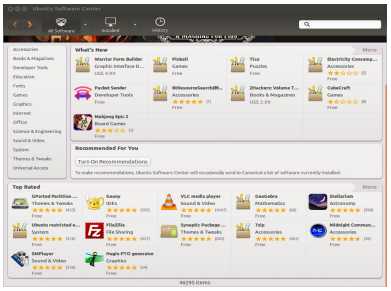
```
user@ubuntu ~ $ apt search kate
```

Package Terminology

- .deb Packages : **Debian, Ubuntu, Mint** and all **derivatives** from Debian and Ubuntu use **.deb** packages.
- .rpm Packages : **Red Hat, Fedora, CentOS**, **OpenSUSE, Mandriva, Red Flag** and **others** use **.rpm** packages.
- dependency : Some packages need **other packages** to function. Tools like apt-get, aptitude and yum will install all dependencies.

Package Terminology

- Open Source Software: Computer software that is released under a license in which the copyright holder **grants users the rights to use**, **study**, **change**, and **distribute** the software and its source code to anyone and for any purpose.[1][2]



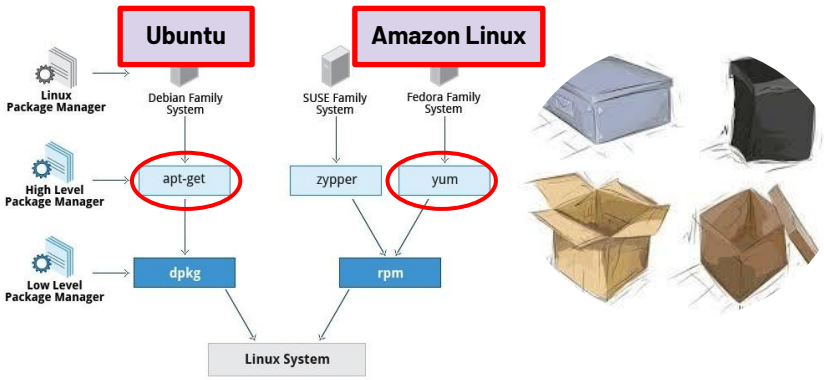
1. "The Open Source Definition". Open Source Org. 7 July 2006. Archived from the original on 11 June 2007. Retrieved 22 January 2020. Open source doesn't just mean access to the source code.
2. ^ "What is Open Source Software". Diffingo Solutions Inc. Archived from the original on 28 October 2008. Retrieved 22 January 2020. Open source software differers from other software because it has a less restrictive license agreement: Instead of using a restrictive license that prevents you from modifying the program or sharing it with friends for example, sharing and modifying open source software is encouraged. Anyone who wishes to do so may distribute, modify or even create derivative works based on that source code!

4

Popular Linux System Package Managers

Popular Linux System Package Managers

Linux systems use package managers to add or remove the software packages. These **package managers are also a package** so you can install any of them.



Popular Linux System Package Managers

Debian Package Managers

dpkg is the **main package management program** for the **Debian Linux** distros. It is used to handle Debian package files with the extension of **.deb**

```
$ dpkg -i [package-name]   # Installing a package
$ dpkg -r [package-name]   #Removing a package
$ dpkg -l                  # Lists installed packages
```

Popular Linux System Package Managers

Debian Package Managers

APT
(Advanced Package Tool)

- The **A**dvanced **P**ackaging **T**ool is what Ubuntu Software Center is built on

- 'apt-get install PACKAGE' will install and organize software
- 'apt-cache list PACKAGE' will search for PACKAGE in the local database
- 'apt-get update' update the local package database

```
$ apt update                    # Update the installed packages
$ apt install [package-name]    # Install a package and all its dependencies
$ apt remove [package-name]     # Remove a package
$ apt purge [package-name]      # Remove a package and its configuration files
```


Popular Linux System Package Managers

Debian Package Managers

Aptitude Package Manager

aptitude tool provides the functionality of **apt-get**, as well as many additional features:

- aptitude provides easy access to all versions of a package
- aptitude tracks of obsolete software
- aptitude has a powerful system for searching particular packages


```
$ aptitude install [package-name]  # Install a package
$ apt-get install [package-name]   # Install a package
```

https://www.tecmint.com/difference-between-apt-and-aptitude/

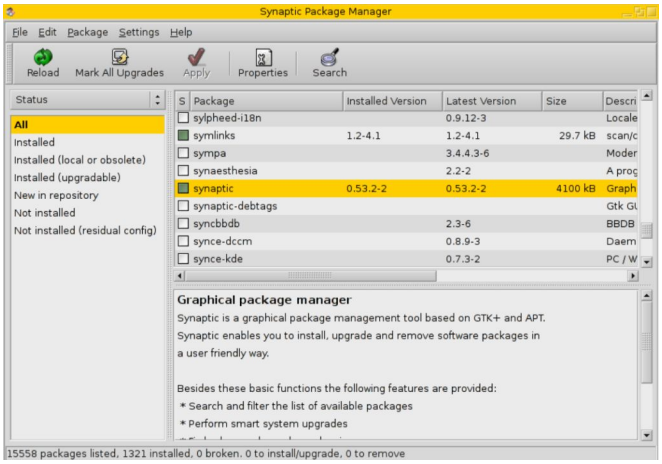
Popular Linux System Package Managers

Debian Package Managers

Synaptic Package Manager

Synaptic is a graphical package manager and used for installing, upgrading and removing single and multiple packages in a more user-friendly way.

```
sudo apt-get install synaptic
```



Popular Linux System Package Managers

Red Hat Package Managers

rpm is the package manager for **Red Hat Linux** operating systems. The installation package files have **.rpm** extension. These files are used for installing programs. **rpm** command has been used for RPM packages by default but new tools are developed for better performance.

```
$ rpm -i [package-name]   # Install a package
$ rpm -e [package-name]   # Uninstall a package
```

Popular Linux System Package Managers

Red Hat Package Managers

YUM (Yellowdog Updater Modified)

YUM is an open-source package manager that was **developed by Duke University**. It is **used** both **in** the **command line and GUI**. It works mostly the same as APT in Debian Linux systems. Here are some examples of YUM.

```
$ yum install [package-name]     # Install a package
$ yum remove [package-name]   # Remove a package
$ yum update [package-name]    # Update a package
```


CLARUSWAY
WAY TO REINVENT YOURSELF

Break
return @ 9pm

Comparing RPM & YUM

Parameter	Yum	RPM
Type	High-level packet manager	Low-level packet manager
Dependencies	Resolves and installs dependencies automatically.	Does not resolve dependencies.
Installation	Can install packages with just the software name (meta-data)	Need to provide exact rpm location & filename
Upgrades	Can perform automatic upgrade to latest version	Upgrades not supported directly

Popular Linux System Package Managers

Red Hat Package Managers

DNF – Dandified Yum

It is the **new generation of YUM** package manager. It is the default package manager of **Fedora 22 and newer** distros. The usage of DNF is mostly the same as YUM.

```
$ yum install dnf    # Install DNF via yum.
$ dnf –version       # Checking DNF version
$ dnf install        # Installing a package
```

Popular Linux System Package Managers

Red Hat Package Managers

Other RPM tools:

- zypper (openSUSE)
- up2date (Red Hat Enterprise Linux, CentOS 3 and 4, and Oracle Linux)
- urpmi (Mandriva Linux, ROSA Linux, and Mageia
- apt-rpm (Ark Linux,[11] PCLinuxOS and ALT Linux)
- smart (Unity Linux and Fedora)
- rpmquery (Red Hat Enterprise Linux)

Popular Linux System Package Managers

Other Package Managers

Below are a few more notable/interesting package managers.

- Portage: Package manager for Gentoo.
- Pacman: Arch Linux Package manager.
- Nix: A 'Fully Functional/Transactional' package manager.

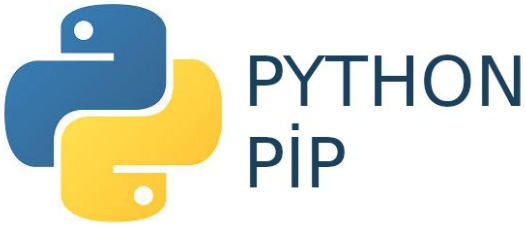
•Brew: An Open Source package manager for OSX.
•Chocolatey: A package manager for Windows.

Popular Linux System Package Managers

Other Package Managers

Programming languages have their own default package managers. They help to find and install the packages via searching libraries that exist on the internet for that language.

Examples: **Python: pip** / Ruby: gem, rubygems / Haskell: cabal / NodeJS: npm



Summary of Commands - yum

Command	Description
`yum install [package-name]`	Install a package
`yum -y install [package-name]`	Skip confirmations during installation
`yum remove [package-name] (also yum erase)`	Remove a package
`yum autoremove [package-name]`	Remove a package and unused dependencies
`yum update [package-name]`	Update a package
`yum update`	Update all installed packages
`yum info [package-name]`	Get information about a package
`yum list`	List all installed and available packages
`yum list [package-name]`	List available matching package(s)
`yum list installed`	List installed packages
`yum --showduplicates list [package-name]`	Lists all available versions
`yum install [package-name]-[version]`	Install a specific version
`yum history`	See history of yum activity
`yum repolist all`	Display configured repos

Package Not Available

```
> yum install clarusway-training
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                               | 3.7 kB 00:00:00
No package clarusway-training available.
Error: Nothing to do
>
```

What to do?

Exercise 1

Update **all** installed packages

List all installed packages start with **http**

Find all available packages start with **http**

Install **httpd** if available. (Skip confirmations during installation)

List installed **httpd** package

Remove **httpd**

List installed **httpd** package

Exercise 2

Uninstall **git** with all unused dependencies

Check installed **git**

Find previous available **git** version

Install previous available **git** version

Check installed **git** version

Update **git** to the **latest** version

Check installed **git** version

Examples

```
# search for packages
yum search <package>
dnf search <package>
zypper search <package>
apt-cache search <package>
apt search <package>
pacman -Ss <package>
```

```
# install packages
yum install <package>
dnf install <package>
zypper install <package>
apt-get install <package>
apt install <package>
pacman -S <package>
```

```
# update package database, not required by yum, dnf and zypper
apt-get update
apt update
pacman -Sy
```

```
# update all system packages
yum update
dnf update
zypper update
apt-get upgrade
apt upgrade
pacman -Su
```

```
# remove an installed package
yum remove <package>
dnf remove <package>
apt-get remove <package>
apt remove <package>
pacman -R <package>
pacman -Rs <package>
```

```
# search for the package name containing specific file or folder
yum whatprovides *<binary>
dnf whatprovides *<binary>
zypper what-provides <binary>
zypper search --provides <binary>
apt-file search <binary>
pacman -Fs <binary>
```

Kahoot!

THANKS!

Any questions?