Open Source Community

Users & Groups, Package managers,
Intro to Processes and Services

Agenda:

-Users

-Groups

-Package Managers

-Processes

-Services

-Managing services

User

Super User (Root)

Root is the default user on Linux OS who has access to all commands and files and it’s not recommended to login as a Root to not accidentally delete or move a system Package or file instead we can use command (sudo)

Adding & Deleting users

Adding user

1. Adding new user

```
#sudo useradd [new_account]
```

2.Setting Password for this user

```
#sudo passwd [exist_account]
```

Deleting user

```
#sudo userdel [account_name]
```

Deleting user and all files created by him and his home Folder

```
#sudo userdel  -r  [account_name]
```

User information

```
#id [user_name]
```

```
[username]:[x]:[UID]:[GID]:[Comment]:[Home directory]:[Default shell]
```

Switch user

```
#sudo su [user_name]
```

To return to the original user

```
#exit
```

Groups

Primary & Secondary Groups

Primary Group: is the default group that a user account belongs to. Every user on Linux belongs to a primary group. A user's primary group is usually the group that is recorded in your Linux system's /etc/passwd file. When a Linux user logs into their system, the primary group is usually the default group associated with the logged in account.

Secondary Groups: Once a user has been created with their primary group, they can be added to secondary groups. Linux system users can have a maximum of 15 secondary groups. A Linux system's groups are stored in the /etc/group file.

Creating & Deleting Groups

Creating a group

```
#sudo newgrp [new_group_name]
```

Deleting a group

```
#sudo delgroup [account_name]
```

Add a user to group(s)

```
#sudo usermod –a –G  [group(s)_name]
```

Package Managers

Packages & Repository

Package: is considered to be a collection of files, it can be an application, a program or even documentation. Packages in Linux are stored in repositories where the package manager can easily find, download, and install them.

Repositories: Those packages are made available to the user of a distribution in central repository. Such repositories includes many thousand of packages, each specially built and maintained for the distribution.

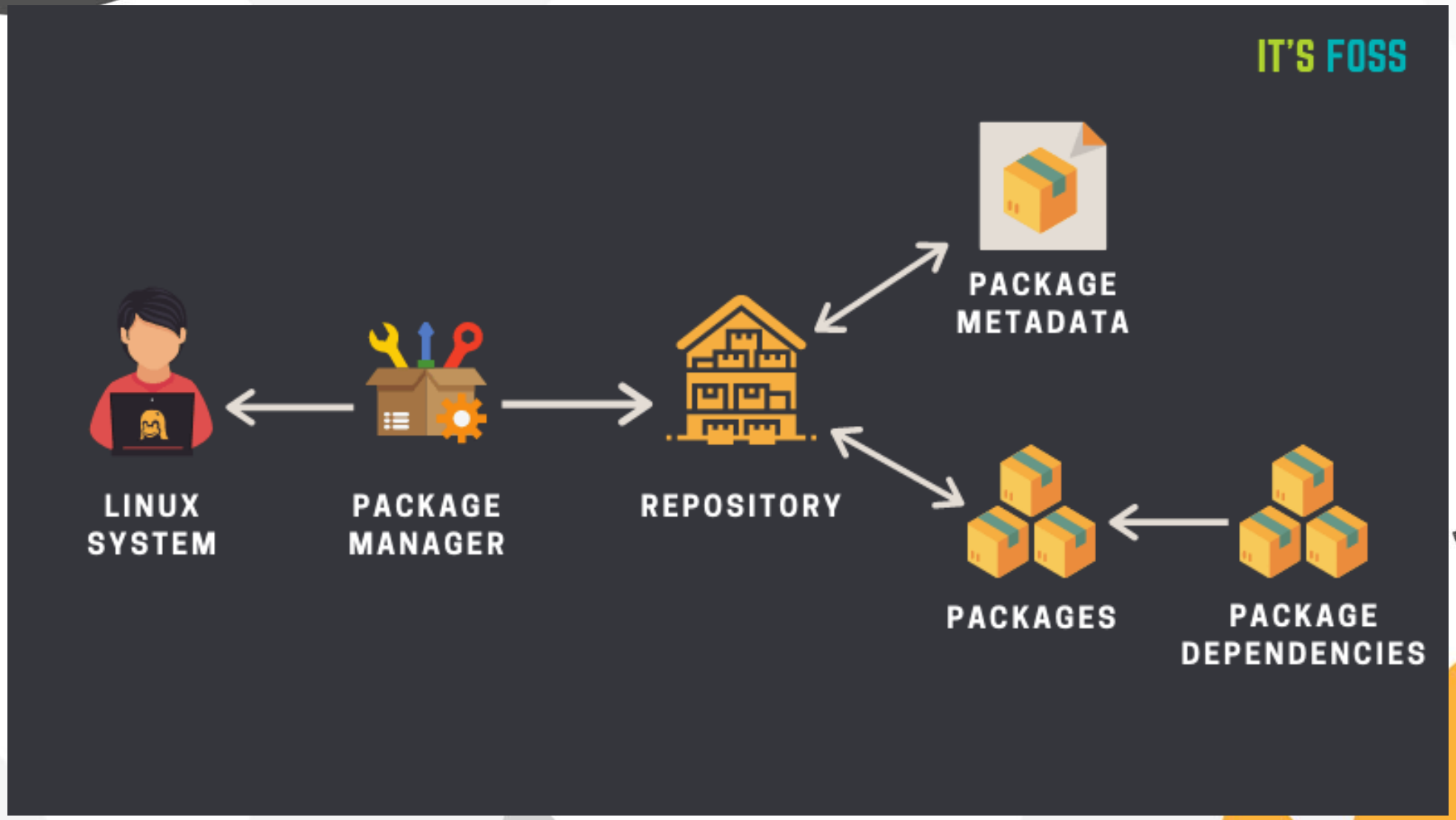
We can consider repositories as an app store.

Dependency: it’s a required package for another package to work like GTK+

Package managers

are a collection of software tools which automate the process of installing, upgrading, configuring and remove software. A package manager maintains a database of information about installed packages (called the **package database**) that enables the package manager to uninstall software, establish whether a new package's dependencies have been met, and determine whether a package you're trying to install has already been installed.

And in debian-based Distributions we have two popular package management tools, one of them is Dpkg which is a low level manager that responsible for the actual installation and compilation of the packages and the other one is APT which is a high level manager and we can say it's the front-end of DPKG and it's responsible for searching the repositories and finding the packages, it is also responsible for resolving dependencies.


IT'S FOSS
PACKAGE METADATA
LINUX SYSTEM
PACKAGE MANAGER
REPOSITORY
PACKAGES
PACKAGE DEPENDENCIES

Installing & Removing Packages

Install a package

```
#sudo apt install [package_name]
```

Remove a Package

```
#sudo apt remove [package_name]
```

Search for packages

```
#sudo apt search [keyword]
```

apt upgrade vs apt update

apt upgrade it upgrades all the packages on your system to their latest versions available in the repositories.

apt update will update the links inside the repository data file so that when you download or update something from the repository you'll get the latest version.

Intro to Processes

Statues of processes

During execution, a process changes from one state to another depending on its environment/circumstances.

In Linux, a process has the following possible states :

. **Running** it's either running or it's ready to run

. **Waiting** in this state, a process is waiting for an event to occur

interruptible -> can be interrupted by signals

uninterruptible -> are waiting directly on hardware conditions and cannot be interrupted by any event/signal.

. **Stopped** in this state, a process has been stopped, usually by receiving a signal.

. **Zombie** here, a process is dead, it has been halted but it’s still has an entry in the process table.

ps command

```
#ps
```

It’s abbreviation to process status

```
#ps aux
```

Show us all of the processes running of the system

PID : process id.
TTY : the type of terminal that the user is logged in to.
Time : time in minutes and seconds that the process has been running.
CMD : The command that launched the process.
you can see man page to see the [options]

top command

#top

It’s abbreviation to process status like ps put it's dynamic real-time view of the running processes

PID: Shows task’s unique process id.
PR: Stands for priority of the task.
SHR: Represents the amount of shared memory used by a task.
VIRT: Total virtual memory used by the task.
USER: User name of owner of task.
%CPU: Represents the CPU usage.
TIME+: CPU Time, the same as ‘TIME’, but reflecting more granularity through hundredths of a second.
SHR: Represents the Shared Memory size (kb) used by a task.
NI: Represents a Nice Value of task. A Negative nice value implies higher priority, and positive Nice value means lower priority.
%MEM: Shows the Memory usage of task.

Signals

The fundamental way of controlling processes in Linux is by sending signals to them. And today we will discuss one of this signals which is KILL which is responsible for ending the chosen process.

we can say it is the twin brother of End Task in windows

```
#kill PID
```

OR

```
#killall [program_name]
```

Service

is a background process that runs without interface by user. This in order to provide even more security, because some of these services are critical for the operation of the operating system.
at sometimes it known as **daemons** and usually these services or daemons names end up with "d". For example, sshd is the name of the service that handles SSH.
And to see all services we will use this command:

```
#sudo systemctl list-unit-files --type service --all
```

Systemd/Init service

Systemd/Init service manger is the mother (parent) of all processes on the system, it’s the first program that is executed when the Linux system boots up; it manages all other processes on the system. It is started by the kernel itself, so in principle it does not have a parent process.

Service status

Enabled Service is enabled at boot time.
Disabled Service is disabled and will not be started at Linux server boot time.
Masked Service is completely disabled and any start operation on it always fails.
Static services will only be used in case another service or unit needs it.
Active Service or daemon is running in the background.

Managing Services

Systemctl command

is the central management tool for controlling the init system / systemd

1. Start a services

```
#sudo systemctl start [service_name]
```

2. Stop a services

```
#sudo systemctl stop [service_name]
```

3. Restart a services

```
#sudo systemctl restart [service_name]
```

4. Check status of a services

```
#sudo systemctl status [service_name]
```

5. enable a services

```
#sudo systemctl enable [service_name]
```

6. disable a services

```
#sudo systemctl disable [service_name]
```

Any Questions?

Thank you

#Stay_Safe