Day 7- Package Management

Linux packages are compressed archives that contain all files needed to install and run a piece of software. They also include metadata like version, dependencies, and installation scripts.

Packages are classified into two main types:

- **RPM** – used by RHEL, CentOS, and Fedora
- **DEB** – used by Ubuntu, Debian, and Linux Mint

Package Managers

A package manager:

- Installs, upgrades, configures, and removes packages
- Verifies digital signatures and checksums
- Manages dependencies
- Groups related packages
- Simplifies software management

Red Hat-Based Distributions

RPM (Red Hat Package Manager)

- Files have the `.rpm` extension.
- Common RPM commands:

```
rpm -ivh <package>      # Install a package
rpm -e <package>        # Uninstall a package
rpm -Uvh <package>      # Upgrade a package
rpm -q <package>        # Query if a package is installed
rpm -Vf <file>          # Verify which package a file belongs to
```

YUM (Yellowdog Updater, Modified)

- YUM is a higher-level tool built on top of RPM.
- It automatically resolves dependencies and uses repositories (local or remote) to manage software.
- Useful YUM commands:

```
yum repolist              # Show all added repositories
yum provides <command>    # Check which package provides a specific command
yum install <package>     # Install a package
yum remove <package>      # Remove a package
yum update <package>      # Update a specific package
yum update                # Update all packages on the system
```

Debian-Based Distributions

Debian-based systems like **Ubuntu**, **Debian**, and **Linux Mint** use the `.deb` package format and tools like `dpkg` and `apt` to manage software.

dpkg (Debian Package Manager)

- `dpkg` is a **low-level** tool used to install, remove, and manage individual `.deb` packages.
- It does not resolve dependencies automatically.
- Common `dpkg` commands:

```
dpkg -i <package>      # Install a .deb package
dpkg -r <package>      # Remove a package
dpkg -l <package>      # List installed package info
dpkg -s <package>      # Show package status and details
dpkg -p <package>      # Show package information
```

apt (Advanced Package Tool)

- `apt` is a **higher-level front end** for `dpkg`.
- It handles dependencies automatically and uses online or local repositories.
- It is more user-friendly and readable compared to older tools like `apt-get` or `apt-cache`.
- APT reads package sources from: `/etc/apt/sources.list`
- Common `apt` commands:

  ```
  apt update                 # Update the package index
  apt upgrade                # Upgrade installed packages
  apt install <package>      # Install a package
  apt remove <package>       # Remove a package
  apt search <package>       # Search for a package
  apt list                   # List packages (installed/available)
  apt edit-sources           # Edit sources.list directly
  ```

- `apt` provides cleaner output and is easier to read than `apt-get`, which is more verbose and script-friendly.
- In older systems, `apt-cache` was used to search for packages (`apt-cache search <package>`), now replaced by `apt search`.