LINUX

Package Management

AUGUST 2025

Installing Software

Package Management Systems on Linux

Package Management Systems on Linux

↘ The core parts of a Linux distribution and most of its add-on software are installed via the Package Management System. Each package contains the files and other instructions needed to make one software component work well and cooperate with the other components that comprise the entire system. Packages can depend on each other. For example, a package for a web-based application written in Python will require the appropriate Python packages to be installed first.

Two Broad Families

↘ There are two broad families of package managers widely deployed: those based on Debian and those which use RPM as their low-level package manager. The two systems are incompatible but, broadly speaking, provide the same essential features and satisfy the same needs. In addition, there are some other systems used by more specialized Linux distributions.

↘ In this section, you will learn how to install, remove, or search for packages from the command line using these two package management systems.

Installing Software

Package Managers: Two Levels

↘ Both package management systems operate on two distinct levels: a low-level tool (such as dpkg or rpm) takes care of the details of unpacking individual packages, running scripts, getting the software installed correctly, while a high-level tool (such as apt, dnf, or zypper) works with groups of packages, downloads packages from the vendor, and figures out dependencies.

↘ Most of the time users need to work only with the high-level tool, which will take care of calling the low-level tool as needed. Dependency resolution is a particularly important feature of the high-level tool, as it handles the details of finding and installing each dependency for you. Be careful, however, as installing a single package could result in many dozens or even hundreds of dependent packages being installed.
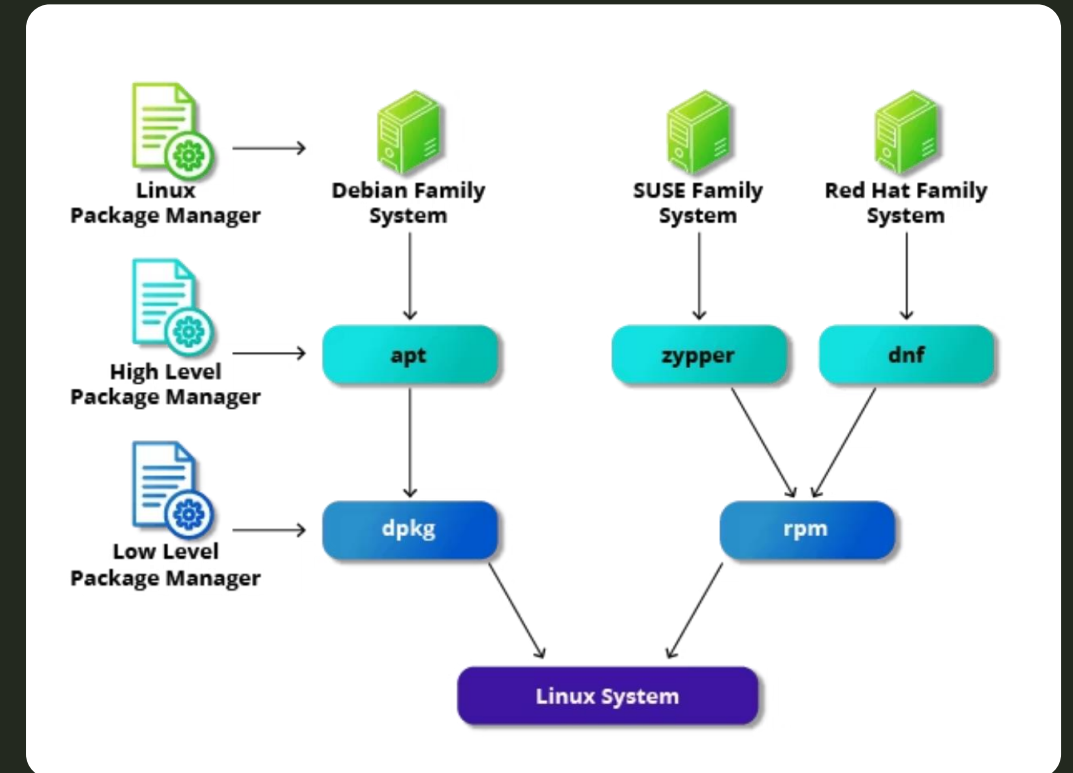
↘ **Low-Level Tools**

dpkg, rpm

Handle individual package installation

↘ **High-Level Tools**

apt, dnf, zypper

Manage dependencies and repositories

Package Managers

## Working With Different Package Management Systems

↘ APT

The Advanced Packaging Tool (apt) is the underlying package management system that manages software on Debian-based systems. While it forms the backend for graphical package managers, such as the Ubuntu Software Center and synaptic, its native user interface is at the command line, with programs that include apt (or apt-get) and apt-cache.

↘ DNF

dnf is the open source command-line package-management utility for the RPM compatible Linux systems that belong to the Red Hat family.

↘ Zypper

zypper is the package management system for the SUSE/openSUSE family and is also based on RPM. zypper also allows you to manage repositories from the command line. zypper is fairly straightforward to use and closely resembles dnf.
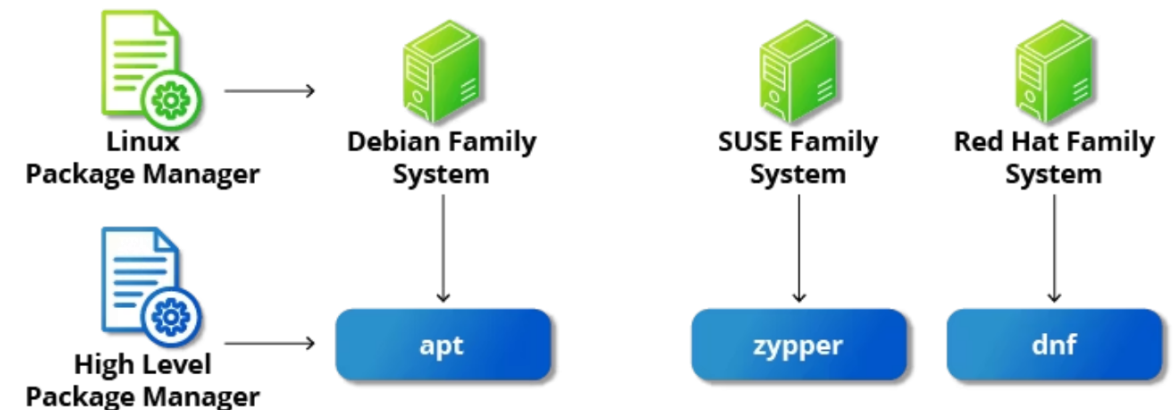
Package Managers

Command Reference

OPERATION	RPM	DEB
Install package	**rpm -i foo.rpm**	**dpkg --install foo.deb**
Install package, dependencies	**dnf install foo**	**apt install foo**
Remove package	**rpm -e foo.rpm**	**dpkg --remove foo.deb**
Remove package, dependencies	**dnf remove foo**	**apt autoremove foo**
Update package	**rpm -U foo.rpm**	**dpkg --install foo.deb**
Update package, dependencies	**dnf update foo**	**apt install foo**
Update entire system	**dnf update**	**apt dist-upgrade**
Show all installed packages	**rpm -qa** or **dnf list installed**	**dpkg --list**
Get information on package	**rpm -qil foo**	**dpkg --listfiles foo**
Show packages named **foo**	**dnf list "foo"**	**apt-cache search foo**
Show all available packages	**dnf list**	**apt-cache dumpavail foo**
What package is **file** part of?	**rpm -qf file**	**dpkg --search file**

↘ This table provides a quick reference for common package management tasks across RPM and Debian-based systems, enabling efficient software management.

info@aistudio.com.tr

aistudio.com.tr

Thank you!