Linux Operating System and Applications

Software Management

Software Types in Linux

Linux software comes in two major forms:

1. RPM Packages

- Precompiled, ready-to-install packages
- Optimized for **Red Hat, Fedora, CentOS**
- Easy to install, update, and remove
- **File format**: `.rpm`

2. Source Code

- Delivered as compressed source files
- Requires **manual compilation**
- More flexible and portable
- Installation is **more complex**

What is RPM?

RPM = Red Hat Package Manager

- Command-line tool for managing `.rpm` packages
- Used in **RHEL, Fedora, CentOS**
- Handles installation, upgrades, queries, and removals **Example**

  RPM File: `nginx-1.24.0-1.el9.x86_64.rpm`

Common RPM Commands

Command	Description	Example
`rpm -i`	Install package	`rpm -i nginx.rpm`
`rpm -U`	Upgrade package	`rpm -U nginx.rpm`
`rpm -e`	Uninstall package	`rpm -e nginx`
`rpm -q`	Query package info	`rpm -q nginx`
`rpm -b`	Build a package	`rpm -ba nginx.spec`

Notice: Always match packages with system architecture (e.g., x86_64 vs i386)

Advanced RPM Queries

Command	Use Case	Example
`rpm -qi`	Info about installed package	`rpm -qi openssh`
`rpm -ql`	List files in a package	`rpm -ql vim`
`rpm -qf <file>`	Which package owns this file?	`rpm -qf /bin/ls`
`rpm -qip <file.rpm>`	Info about an RPM before installing	`rpm -qip nginx.rpm`
`rpm -qc`	Show config files	`rpm -qc httpd`
`rpm -qd`	Show documentation	`rpm -qd bash`
`rpm -qa`	List all installed packages	`rpm -qa | grep nginx`

Dependency Management

Problem: Installing software often requires dependencies.

```
# rpm -ihv MagicPoint-1.09a-1.i386.rpm
error: failed dependencies:
VFlib2 >= 2.25.6-4 is needed by MagicPoint-1.09a-1
libpng.so.2 is needed by MagicPoint-1.09a-1
```

Solution:

- Install dependencies first
- Or use **yum**/**dnf** which handles dependencies automatically

```
# rpm -ihv MagicPoint-1.09a-1.i386.rpm \
            VFlib2-2.25.6-4.i386.rpm \
            libpng-1.0.12-2.i386.rpm
```

Checking System Information

- Use the **uname** command to gather system info:

Command	Purpose
`uname -a`	All system info
`uname -p`	Processor architecture
`uname -i`	Hardware platform
`uname -r`	Kernel release
`uname -v`	Kernel version

Tip: Useful when choosing the right package architecture

Installing from Source (Overview)

Some software is provided as **source code** for flexibility or customization.

Common formats: `.tar.gz`, `.tar.bz2`, `.tgz`

General Steps:

1. Extract the archive
2. Configure the build
3. Compile the code
4. Install the binary

Extracting Source Archives

- Use the tar command

File Type	Command
`.tar`	`tar -xvf file.tar`
`.tar.gz` or `.tgz`	`tar -xvzf file.tar.gz`
`.tar.bz2`	`tar -xvjf file.tar.bz2`
Extract to a folder	`tar -xvzf file.tar.gz -C /opt/myapp`

Configuring Source Code

- Navigate into the extracted directory:

```
cd myapp/
```

```
./configure
```

- Checks for dependencies
- Supports customization

```
./configure --prefix=/usr/local --enable-ssl
```

- To list options:

```
./configure --help
```

Compiling the Code

- Use the ”make” command to compile

```
#make
```

- Requires **GCC** or compatible compiler
- If you modify the source, rerun make

Installing and Uninstalling

- Install

```
#make install
```

  - Copies compiled binaries to system directories (e.g., /usr/local/bin)

- Uninstall / Clean Build Files

```
#make clean
```

⚠️ Not all software supports make uninstall by default.

Installing Software with YUM

YUM = Yellowdog Updater, Modified

- Works with repositories (online software sources)
- Automatically resolves dependencies

Basic Commands:

Command	Description
`yum install nginx`	Install package
`yum update`	Update all packages
`yum update nginx`	Update nginx only
`yum remove nginx`	Remove package
`yum list installed`	View installed packages
`yum search ftp`	Search for packages

YUM Repository Configuration

- Repo config files: `/etc/yum.repos.d/*.repo`

To Add Extra Repositories:

- Example: RPM Fusion Visit: http://rpmfusion.org/Configuration/

💡 Use third-party repos carefully — trust only verified sources

Summary

- Use **RPM** for fast, local package management
- Use **YUM** for dependency-aware online installation
- Use **source code** when you need customization or flexibility
- Always check system architecture and dependencies

Q&A