📦 Linux Package Management Summary

Managing software in Linux is done through **package managers**, which install, update, and remove software packages efficiently.

🔑 Key Features of Package Management

- Downloading and installing packages
- Resolving dependencies
- Standard package formats (`.deb`, `.rpm`, etc.)
- Configuring software
- Managing updates and deletions

📂 Package Types

- **`.deb`** – Debian-based systems (Debian, Ubuntu, Kali, Parrot OS)
- **`.rpm`** – Red Hat-based systems (RHEL, CentOS, Fedora)

📋 Common Package Managers

Command	Description
`dpkg`	Install/remove `.deb` packages (low-level)
`apt`	High-level interface for package management
`aptitude`	Alternative to `apt` with GUI/CLI options
`snap`	Install isolated Snap packages
`pip`	Install Python packages
`gem`	Install Ruby packages
`git`	Clone code repositories (not a package manager but useful for tools)

🚀 APT – Advanced Package Tool (Debian-based distros)

View APT Repositories

```
cat /etc/apt/sources.list.d/parrot.list
```

Search Packages

```

```

```
apt-cache search impacket
```

Show Package Info

```
apt-cache show impacket-scripts
```

List Installed Packages

```
apt list --installed
```

Install Package

```
sudo apt install impacket-scripts -y
```

🔃 GIT – Cloning from GitHub

Create a folder and clone:

```
mkdir ~/nishang/ && git clone https://github.com/samratashok/nishang.git ~/nishang
```

📥 Using `dpkg` to Install `.deb` Packages

Step 1: Download a `.deb` file

```
wget http://archive.ubuntu.com/ubuntu/pool/main/s/strace/strace_4.21-1ubuntu1_amd64.deb
```

Step 2: Install using `dpkg`

```
sudo dpkg -i strace_4.21-1ubuntu1_amd64.deb
```

🧪 Test the Installation

```
strace -h
```

✅ Summary

- Use `apt` for easy installs with dependency handling.
- Use `dpkg` for manual `.deb` installs.
- Use `apt-cache` and `apt list` to search or list packages.
- Use `git`, `pip`, or `gem` for language-specific or code-based installations.

Pro Tip: Always update the APT cache before installing packages.

```
sudo apt update
```

Experiment safely in a VM to understand the behavior of each tool.