

## Exercise 1: Query an installed package

1. List all installed packages:

```
sudo dpkg -l
```

2. Find the full package name for bash:

```
sudo dpkg -l bash
```

or

```
apt list --installed bash
```

3. Get detailed info about the bash package:

```
apt show bash
```

4. List all files installed by the bash package:

```
sudo dpkg -L bash
```

## Exercise 2: Install and remove using apt and dpkg

Use a package like `ksh` (or any package that exists in Debian).

1. Download a `.deb` package (without installing):

```
apt-get download ksh
```

This saves `ksh_*.deb` in the current directory.

2. Download the `.deb` into a specific directory, e.g. `/tmp`:

```
cd /tmp
```

```
apt-get download ksh
```

3. Install the .deb file with dpkg:

```
sudo dpkg -i /tmp/ksh_*.deb
```

If there are missing dependencies, fix them with:

```
sudo apt -f install
```

4. Use dpkg/apt to install if it does not exist and upgrade if it does:

- With dpkg, you just run the same command again:

```
sudo dpkg -i /tmp/ksh_*.deb
```

- With apt, you typically upgrade like this:

```
sudo apt install --only-upgrade ksh
```

or simply:

```
sudo apt install ksh (it will upgrade if a newer version is available).
```

### Exercise 3: Searching and showing package info

1. Search for a package containing “editor”:

```
apt search editor
```

or specifically for nano:

```
apt search nano
```

2. Get detailed info about a package:

- apt show nano

- apt show vim

3. Optional “what provides this file” (similar to whatprovides, needs extra tool):

```
sudo apt install apt-file
```

```
sudo apt-file update
```

```
apt-file search /usr/bin/vim
```

4. List dependencies of nano:

```
apt-cache depends nano
```

or on newer systems:

```
apt depends nano
```

## Exercise 4: Working with groups (tasks / meta-packages)

Debian uses “tasks” and meta-packages instead of DNF groups.

1. List available tasks using tasksel:

```
sudo apt install tasksel
```

```
sudo tasksel --list-tasks
```

2. Install a task (for example “standard”) or a development meta-package:

- Task:

```
sudo tasksel install standard
```

- Meta-package (similar to “Development Tools”):

```
sudo apt install build-essential
```

3. Remove a task or meta-package:

- Task:

```
sudo tasksel remove standard
```

- Meta-package:

```
sudo apt remove build-essential
```

## Exercise 5: Update system (and remove obsolete packages)

1. Update package lists:

```
sudo apt update
```

2. Upgrade all packages:

```
sudo apt upgrade
```

For a full distribution upgrade (may add/remove packages):

```
sudo apt full-upgrade
```

3. Remove no-longer-needed automatically installed packages:

```
sudo apt autoremove
```

## Exercise 6: Clean APT cache

1. Clear all cached packages:

```
sudo apt clean
```

(Removes downloaded .deb files from `/var/cache/apt/archives`.)

2. Download ksh and store it in the default cache (download only, no install):

```
sudo apt-get install --download-only -y ksh
```

```
ls /var/cache/apt/archives | grep ksh
```

3. Install ksh from the cache (or from a copied .deb):

```
cd /var/cache/apt/archives
```

```
sudo dpkg -i ksh_*.deb
```

- 1.