

# LPIC-1 101-500 – Lesson 15

## 102.4 Use Debian package management



# Introduction

- Linux systems use package manager for the installation of software, documentation, libraries, etc
- This makes the use of free/open source software very convenient for Linux users because they do not have to compile the program from source
- Package Managers use **repositories** for saving the packages and the program code

Package Management is the single biggest advancement  
Linux has brought to the industry

~ Ian Murdock ~




# Functions of Package Managers

- Install software from repositories
- Verify digital signatures and checksums to ensure that the software has not been tampered with
- Upgrade software to newer editions
- Packaging of recently released software
- Dependency resolution
- Categorization of Packages



# The Debian Package Manager

- **Debian** systems and derivatives use the **dpkg** package manager
  - Debian package files have a **.deb** extension
  - Packages can be initialized during the installation using a TUI
  - The **apt** utility is used for downloading packages from repositories and for dependency resolution
  - **dkpg**: Used for the installation of packages through **.deb** files, for uninstalling software and many other functions
  - **dpkg-reconfigure**: is used to reconfigure an already installed package
- 

# Advance Packaging Tool - APT

- **APT** is in charge of package downloading, installing/uninstalling packages, dependency resolution, showing package information, etc
- Used **dpkg** in the backend for installation/removal of packages
- **apt-get**: used for installation/removal/upgrade of packages and system update/upgrade
- **apt-cache**: this command is used for package queries and showing package information
- The **apt** configuration files are under the **/etc/apt** directory. One of the most useful configuration file is **/etc/apt/sources.list** (main repository configuration) and the **/etc/apt/sources.list.d/\*.list** (custom repository configuration) where we define the repositories used by **apt**

# Sample */etc/apt/sources.list* file

## # Main Repository

```
deb http://deb.debian.org/debian/ buster main non-free contrib  
deb-src http://deb.debian.org/debian/ buster main non-free contrib
```

## # Security updates

```
deb http://security.debian.org/debian-security buster/updates main contrib non-free  
deb-src http://security.debian.org/debian-security buster/updates main contrib non-free
```

## # buster-updates, previously known as 'volatile' (frequently updated software)

```
deb http://deb.debian.org/debian/ buster-updates main contrib non-free  
deb-src http://deb.debian.org/debian/ buster-updates main contrib non-free
```

## # buster-backports, previously on backports.debian.org (new/experimental software)

```
deb http://deb.debian.org/debian/ buster-backports main contrib non-free  
deb-src http://deb.debian.org/debian/ buster-backports main contrib non-free
```



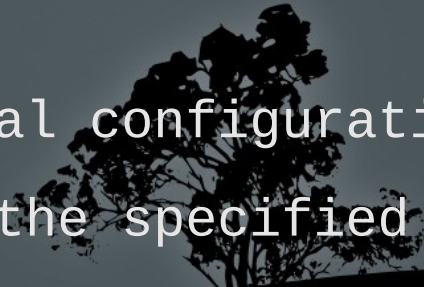
# The `dpkg` command

- `# dpkg -i apache2_2.2.16-6+squeeze4_i386.deb`  
# install apache2 package from a .deb file
- `# dpkg -r apache2` # uninstall apache2 package but keep the configuration files
- `# dpkg --purge apache2` # uninstall apache2 package and delete the configuration files



# The `dpkg` command


## Options:

- `-i, --install` # install package from `.deb` file
  - `-r, --remove` # remove package and keep the configuration files
  - `-P, --purge` # remove package along with the configuration files
  - `-s, --status` # show package status and information
  - `-S, --search` # search for keywords in all the installed packages
  - `-L, --listfiles` # show all files in a package file
  - `-l, --list` # show information about packages that match the query
  - `--unpack` # install package without initial configuration
  - `-x --extract` # install package files in the specified directory
- 



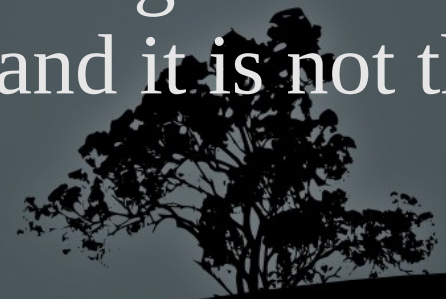
# The `dpkg` command

## Options:

- `-p, --print-avail` # print information from installed package
  - `-I, --info` # print information from `.deb` file
  - `--configure` # configure an incompletely installed package
  - `-C, --audit` # search for incompletely installed packages and advise about possible resolution steps
  - The `dpkg` data directory is under `/var/lib/dpkg`. Here we can find information about installed packages
- 

# The `dpkg-reconfigure` command

- The **dpkg-reconfigure** command is used to reconfigure already installed packages. If the package supports a TUI this will be invoked
- **# dpkg-reconfigure postfix #**  
reconfigure the **postfix** package
- **# dpkg-reconfigure locales #**  
reconfigure the **locales** package
- The **dpkg --configure** is for configuring incompletely installed packages and it is not the same as **dpkg-reconfigure**



# The `apt-get` command

- The **apt-get** command is used for the installation of packages from the repositories, resolution of dependencies and conflicts, package removal and system upgrade
- The **apt-get** command is invoked as follows:
  - **apt-get <options> [commands]**
  - Options can be single dashed (-) or double dashed (--) and they are optional
  - Commands do not have a dash and they are obligatory




# Install packages with `apt-get`

- `# apt-get install apache2` # install the `apache2` package and dependencies. You should respond with "y" or simply "Enter" for the installation to proceed. If the package exists it will be upgraded to the most recent version/revision
- `# apt-get -y install vim` # install package `vim` with "Yes" as a selected option. If the package exists it will be upgraded to the most recent version/revision



# Remove and purge packages with `apt-get`

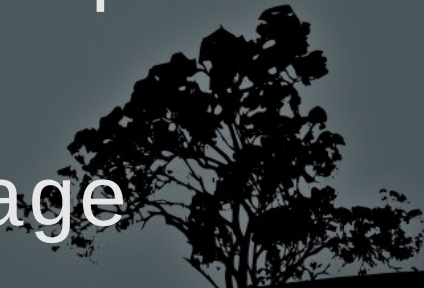
- `# apt-get remove apache2 # remove package apache2 while keeping its configuration`
  - `# apt-get --purge remove apache2 # = apt-get purge apache2 remove package apache2 deleting its configuration`
  - `# apt-get clean # clean the temporary apt cache /var/cache/apt from downloaded .deb files`
- 

# Update/Upgrade system with `apt-get`


- `# apt-get update` # retrieve updated information about the new version/revisions of installed packages
- `# apt-get upgrade` # upgrade all packages for which new versions/revisions are available
- `# apt-get dist-upgrade` # upgrades all packages for which new versions/revisions are available but using a smarter conflict resolution algorithm than `apt-get upgrade`

# Some more options of `apt-get`

## Options:

- `-d` # download .deb files from repositories but without installing them
  - `-s` # simulate all the steps without actually installing or removing
  - `-y` # answer "Yes" to all questions
  - `--purge` # purge (remove configuration) of package
- 

# The `apt-cache` command

- `$ apt-cache search # search for regular expressions in the package names or descriptions`
  - `$ apt-cache show apache2 # information about the apache2 package`
  - `$ apt-cache showpkg apache2 # provides alternative information like possible other versions, dependencies, reverse dependencies etc`
  - `$ apt-cache depends apache2 # show apache2 package dependencies`
  - `$ apt-cache rdepends apache2 # show apache2 package reverse dependencies, i.e. packages that depend from it`
  - `$ apt-cache stats # statistics about installed packages`
- 



# The `apt` command

- The **apt** command combines the commands of **apt-get**, **apt-cache** and other APT commands. The goal is for **apt** to replace the other commands in the future.
- `# apt install apache2 # install apache2 package`
- `# apt remove apache2 # remove package apache2 keeping its configuration`
- `# apt purge apache2 # remove package apache2 deleting its configuration`
- `# aptitude clean # = apt-get clean`

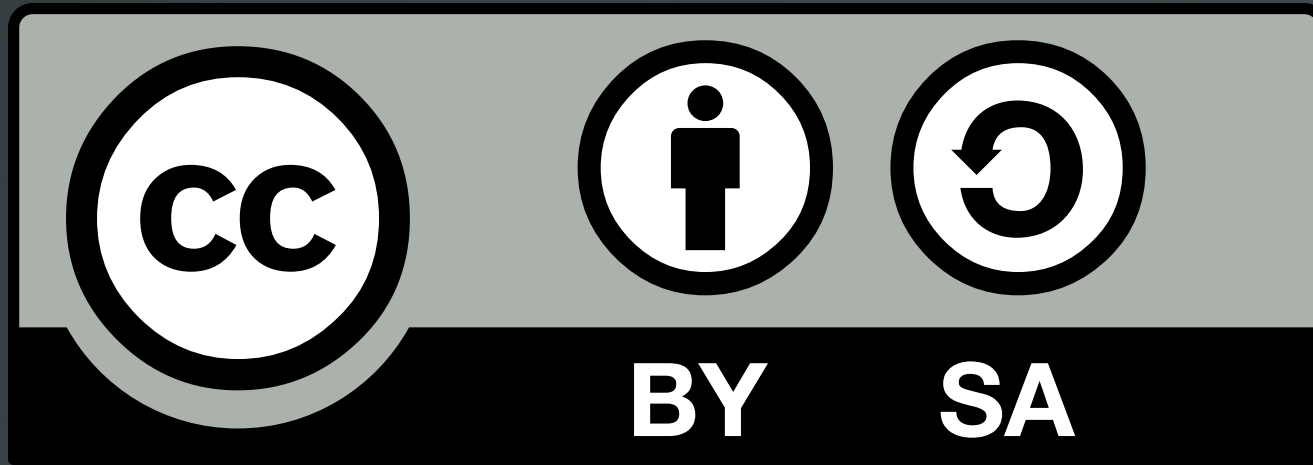


# The `aptitude` command

- `# apt search keyword` # similar to `apt-cache search` but searched only filenames by default
- `# apt update` # = **apt-get update**
- `# apt safe-upgrade` # = **apt-get upgrade**
- `# apt full-upgrade` # = **apt-get dist-upgrade**
- `# apt show apache2` # show package information (similar to `apt-cache show`)



# License



The work titled "LPIC-1 101-500 – Lesson 15" by Theodotos Andreou is distributed with the Creative Commons Attribution ShareAlike 4.0 International License.

