Debian Administration

# Session 1

Sunday, December 2, 2013

## Part I: Debian System Installation

To use Debian, you need to install it on a computer; this task is taken care of by the debian-installer program. A proper installation involves many operations. This part reviews them in their chronological order.

### Reference:

*The Debian Administrator’s Handbook, Chapter 4*

### Key Knowledge Areas

1. Debian Installation Methods
2. Installing Debian Step by Step
3. Debian after First Boot

Debian Installation requires 56 MB of RAM (Random Access Memory) and at least 650 MB of hard drive space. All Falcot computers meet these criteria. Note, however, that these figures apply to the installation of a very limited system without a graphical desktop. A minimum of 512 MB of RAM and 5 GB of hard drive space are really recommended for a basic office desktop workstation.

#### Debian Installation Methods:

A Debian system can be installed from several types of media, as long as the BIOS of the machine allows it. You can for instance boot with a CD-ROM, a USB key, or even through a network.

1. **Installing from a CD-ROM/DVD-ROM**: The most widely used installation media is CD-ROM (or DVD-ROM, which behaves exactly the same way): the computer is booted from this media, and the installation program takes over.
2. **Booting from a USB Key**: You must first identify the peripheral name of the USB key (ex: /dev/sdb); the simplest means to do this is to check the messages issued by the kernel using the dmesg command. Then you must copy the previously downloaded ISO image (for example debian-6.0.0-amd64-i386-netinst.iso) with the command

cat debian-6.0.0-amd64-i386-netinst.iso >/dev/sdb;sync

This command requires administrator rights, since it directly accesses the USB key peripheral and blindly erases its content.

1. **Installing through Network Booting**: Many BIOSes allow booting directly from the network by downloading the kernel to boot. This method (which has several names, such as PXE or TFTP boot) can be a life-saver if the computer does not have a CD-ROM reader, or if the BIOS can't boot from such media.
2. **Other Installation Methods, FAI**: Depending on the situation and the complexity of the installations to be made, we can use FAI (Fully Automatic Installer, described in Section 12.3.1, “Fully Automatic Installer (FAI)” (page 337) from the reference), or even a customized installation CD with preseeding (see Section 12.3.2, “Preseeding Debian-Installer” (page 338) from the reference).

#### Installing Debian Step by Step

For a standard installation, you only need to choose “Install” or “Graphical install” (with the arrows), then press the Enter key to initiate the remainder of the installation process.

Installation alongside an existing Windows system:

If the computer is already running Windows, it is not necessary to delete the system in order to install Debian. You can have both systems at once, each installed on a separate disk or partition, and choose which to start when booting the computer. This configuration is often called “dual boot”, and the Debian installation system can set it up. This is done during the hard drive partitioning stage of installation and while setting up the bootloader.

If you already have a working Windows system, you can even do without the recovery CD-ROM; Debian offers a Windows program that will download a light Debian installer and set it up on the hard disk. You then only need to reboot the computer and choose between normal Windows boot or booting the installation program. You can also find it on a dedicated website with a rather explicit name…

🡺 http://ftp.debian.org/debian/tools/win32-loader/stable/

🡺 http://www.goodbye-microsoft.com/

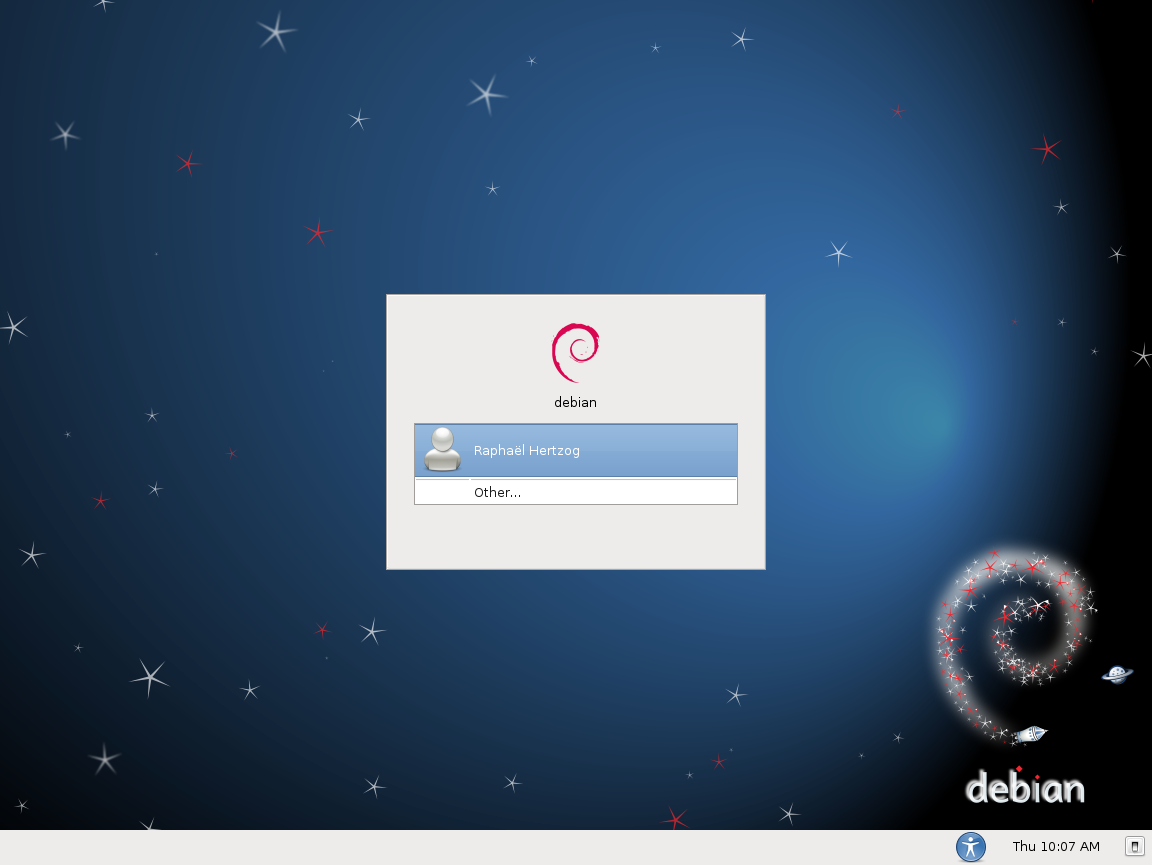
After CD/DVD insert, the Debian installation menu appear, each menu entry hides a specific boot command line, which can be configured as needed by pressing the TAB key before validating the entry and booting. The “Help” menu entry displays the old command line interface, where the F1 to F10 keys display different help screens detailing the various options available at the prompt. You will rarely need to use this option except in very specific cases.



* Selecting the language
* Selecting the country
* Selecting the keyboard layout
* Detecting Hardware
* Loading Components
* Detecting Network Hardware
* Configuring the Network
* Configuring the Clock
* Administrator Password
* Creating the First User
* Detecting Disks and Other Devices
* Starting the Partitioning Tool
* Installing the Base System
* Configuring the Package Manager (apt)
* Debian Package Popularity Contest
* Selecting Packages for Installation
* Installing the GRUB Bootloader
* Finishing the Installation and Rebooting

#### Debian After the First Boot

If you activated the task “Graphical desktop environment”, the computer will display the **gdm** login manager.



Installing Additional Software:

* The installed packages correspond to the profiles selected during installation, but not necessarily to the use that will actually be made of the machine.
* To facilitate the installation of coherent groups of programs, Debian creates “tasks” that are dedicated to specific uses (mail server, file server, etc.). You already had the opportunity to select them during installation, and you can access them again thanks to package management tools such as **aptitude** (the tasks are listed in a distinct section) and **synaptic** (through the menu Edit 🡪 Mark Packages by Task…)

## Part II: Debian Package Management

*Candidates should be able to perform package management using the Debian package manager.*

### Key Knowledge Areas

* Install, upgrade and uninstall Debian binary packages.
* Find packages containing specific files or libraries that may or may not be installed.
* Obtain package information like version, content, dependencies, package integrity and installation status (whether or not the package is installed).

#### Introduction

Most Linux distributions manage software using some form of package management to perform tasks such as installations, updates and queries. There are two main package management system in use today and what your distribution uses will depend on its heritage. Most distributions derived from Redhat use the rpm package manager, while those that are derived from Debian use the dpkg manager.

**Debian Package Management**

Systems using Debian based variants of Linux use the Debian Package Management system. The Debian system is more rigorous and configurable than the rpm system and is used by Debian derivatives such as Debian. Under Debian system's packages are managed with '**dpkg**' but you may be more familiar with dpkg management tools such as “**apt**” and “**aptitude**”.

***Package Naming***

Debian package names are formed as follows:

*name\_version-release\_architecture.deb*

*e.g.* xxchat\_2.8.6-4debian5\_amd64.deb

The release number indicates which Debian release of the version of the software the package contains, while the architecture name specifies the computer architecture (i386, sparc, all). So the above deb package is for xchat version 2.8.6-4debian5 for the AMD64 architecture.

**dpkg**

The **dpkg** command is controlled via command line parameters, which consist of an action and zero or more options. The action parameter tells dpkg what to do and options control the behaviour of the action in some way.

**dpkg** maintains some usable information about available packages. The information is divided in three classes: **states**, **selection states** and **flags**.

**Package States**

|  |  |
| --- | --- |
| ***State*** | ***Description*** |
| **installed** | The package is unpacked and configured OK. |
| **half-installed** | The installation of the package has been started, but not completed for some reason. |
| **not-installed** | The package is not installed on your system. |
| **unpacked** | The package is unpacked, but not configured. |
| **half-configured** | The package is unpacked and configuration has been started, but not yet completed for some reason. |
| **config-files** | Only the configuration files of the package exist on the system. |

**Package Flags**

|  |  |
| --- | --- |
| ***Flag*** | ***Description*** |
| **hold** | A package marked to be on **hold** is not handled by **dpkg**, unless forced to do that with option --**force-hold**. |
| **reinst-required** | A package marked **reinst-required** is broken and requires reinstallation. These packages cannot be removed, unless forced with option **--force-reinstreq**. |

**Actions**

The heart of dpkg operation is the command line parameters specifying the action which should be performed. While there are a large number of these, the following table summarises the main actions you are likely to require on any regular basis.

|  |  |
| --- | --- |
| ***Action*** | ***Description*** |
| **-l** | Prints a list of the packages installed on the system, or matching a pattern if any is given. The first three characters on each line show the state, selection state, and flags of the package |
| **-s** | Shows the status and information about particular installed package(s) |
| **-I** | Show information about a package in a **.deb** file |
| **-L** | List the files included in a package |
| **-S** | Show the package which includes the file specified |
| **-i** | Install (or upgrade) and configure a package from a **.deb** file |
| **--unpack** | Unpack (only) a package in a **.deb** file |
| **--configure** | Configure an unpacked package. With **-a** (or **--pending)** configures all packages requiring configuration |
| **-r** | Remove a package (but leave its configuration files) |
| **-P** | Purge – remove a package along with its configuration files |
| **--get-selections** | Get a list of package selections from a system (to stdout) |
| **--set-selections** | Set the list of package selections for a system (from stdin) |

**Options**

All options can be specified both on the command line and in the **dpkg** configuration file */etc/dpkg/dpkg.cfg*. Each line in the configuration file is either an option (exactly the same as the command line option but without leading dashes) or a comment (if it starts with a **#**).

|  |  |
| --- | --- |
| ***Option*** | ***Description*** |
| **--force-*thing*** | Forces **dpkg** to perform an action which it would normally not take (for example, to ignore dependency information - **--force-depends**, or to downgrade a package with –**force-downgrade**) |
| **--refuse-*thing*** | Refuse to do something which **dpkg** would normally automatically do |
| **--ignore-depends** | Ignore dependency checking for a package |
| **--no-act** | Show what **dpkg** would do, but don't do it (also: **--simulate)** |
| **-R** | Recurse through directories (using with **-i** or **--unpack**) |

***Files***

**dpkg**

**dpkg** uses a number of files in its operation, including **/etc/dpkg/dpkg.cfg** which contains default configuration settings.

Lists of available packages along with their statuses are held in the files

**/var/lib/dpkg/available** and **/var/lib/dpkg/status**.

A **.deb** file, along with the files making up a packages programs, libraries and configuration, will also include a number of control files which allow the execution of scripts before and after installation and removal, along with lists of files and configuration files. These can be found in the **/var/lib/dpkg/info** directory once the packages are installed.

**Use of *dpkg***

To install a package from a .deb file, you could use dpkg as follows:

|  |
| --- |
| # dpkg –i hello\_2.1.1-4\_i386.deb OR  # dpkg --unpack hello\_2.1.1-4\_i386.deb  # dpkg --configure hello |

To remove the hello package along with its configuration, you could use:

|  |
| --- |
| # dpkg –P hello |

While:

|  |
| --- |
| # dpkg –r hello |

would remove only the package, leaving its configuration files installed.

The get a list of all the packages installed on the system, use the command:

|  |
| --- |
| # dpkg –l |

Note that when dealing with a package file, the filename is given, while when dealing with an installed package, only the package name is given.

***APT***

The **dpkg** tool is fine for installing individual packages with no dependencies, but when installing a number of packages which may have dependencies, the APT tool is generally used instead.

APT is one of the strengths of dpkg, and provides an easy way of installing and updating a system. It is controlled by two files:

|  |  |
| --- | --- |
| ***File*** | ***Description*** |
| **/etc/apt/apt.conf** | Contains general configuration options for APT, such as which release of Debian to install, whether/which proxy settings to use, etc |
| **/etc/apt/sources** | Lists sources of Debian files, which may be on CDs, or on the network |

In general, to use APT you must first configure the sources it is to used. The main configuration file for apt is the /etc/apt/source.list file. This defines the repositories apt should use for installing new software or for updating existing applications. Below is an excerpt from an Debian /etc/apt/sources.list file.

deb http://archive.debian.com/debian squeez squeez main restricted

deb-src http://archive.debian.com/debian squeez main restricted

deb http://archive.debian.com/debian squeez-updates main restricted

deb-src http://archive.debian.com/debian squeez-updates main restricted

The type of archive can be either deb for binary packages or deb-src for source files. The rest of the line define repository location such by providing the url at which the repository can be found, and details needed to find the specific repository at the url location.

Once APT knows where the Debian packages are located, two command line tools are used for package management: **apt-cache** and **apt-get**.

**apt-cache**

**apt-cache** allows manipulation of the APT package cache (which is stored in files in **/var/cache/apt**). An action normally follows apt-cache on the command line, and common options include:

|  |  |
| --- | --- |
| ***Action*** | ***Description*** |
| **search** | Search all the available package descriptions for the string given, and print a short description of the matching package |
| **show** | Shows a full description of the package specified |

**apt-get**

While **apt-cache** is useful for finding out information about available packages, **apt-get** allows updating of package information, retrieval, installation and removal of packages, and even upgrading of an entire Debian distribution. apt-get expects an action to be provided on the command line, and the most common are listed below:

|  |  |
| --- | --- |
| ***Action*** | ***Description*** |
| **update** | Update the list of packages from the sources in /etc/apt/sources.list |
| **install *package*** | Install the package(s) specified, along with any dependencies |
| **upgrade** | Upgrade any packages which have newer versions available |
| **dist-upgrade** | Upgrade entire distribution to the latest release (best to read the release notes first!) |
| **remove** | Remove the package(s) specified |

**Use of APT**

One use of APT is for updating the system (for example if security-related updates have become available). This is normally done using the two commands:

|  |
| --- |
| # apt-get update  # apt-get upgrade |

The other main use of APT is to install required packages. This normally involves the following commands:

apt-get update #update list of packages

apt-cache search frob #find packages relating to frobbing

apt-cache show frobnicate #show information regarding a particular package

apt-get install frobnicate #install frobnicate package and its dependencies

***The Alien Tool***

The **alien** tool will change Debian packages into RedHat ones and vice versa. One can download it at: [http://kitenet.net/programs/](http://kitenet.net/programs/" \t "_top)

Convert a debian package to an rpm:

|  |
| --- |
| #alien --to-rpm package.deb |

Convert an rpm package to debian:

|  |
| --- |
| # alien --to-debian package.rpm |

**Aptitude**

Aptitude is another text based front-end to the Debian package management system. Much like apt, aptitude can be used to install, remove and update software on Debian machines. Below is a list of commonly used parameters with aptitude:

* aptitude update – update the list of available packages.
* aptitude install – will install new software ,
* aptitude reinstall – will reinstall an existing package
* aptitude remove – will remove an existing package
* aptitude purge – will remove a package and its associated configuration files
* aptitude search – will allow you to search the list of available packages.

Used files, terms and utilities:

* /etc/apt/sources.list
* dpkg
* dpkg-reconfigure
* apt-get
* apt-cache
* aptitude