System Administration for Beginners

Week 6 Homework

October 20, 2008

1 Introduction

In lecture and lab, we've explored some of the functionality of Debian's package management system, **APT**, through the the use of it's tools, namely `apt-get`. Let's explore a bit deeper into **APT** and learn more about what it can do for us.

2 Submission

As usual, please submit your homework to us at `cardi+decal@ocf.berkeley.edu` and `jchu+decal@ocf.berkeley.edu`. All that is necessary in your submission are your names, `inst` logins, project group number, and the answers. Please read each section of this homework carefully to understand what needs to be submitted.

3 Homework

Most GNU/Linux distributions have several "releases", depending on the stability and maturity of packages. For example, one release may be distributed for stability and long term support, meaning that it would be ideal for a production server. Another release may have the latest and greatest version of each software package, but updates and fixes are continuously being released.

1. How many "releases" of Debian are there, what are they called, and what are their codenames?

2. What "release" of Debian are we using? How can we tell?

With the Debian release being used, we are not always downloading and installing the latest packages. It is possible, however, through the tools that APT provides, to upgrade our Debian release. In order to do so, we need look into the directory `/etc/apt` and change certain values. Like apache2's configuration file, the configuration for APT has it's own format and options (look at the man page for `apt.conf`).

3. Inside the configuration directory, there's a file that contains the URL for the package repository. What is the name of the file?

4. The sources being used have a specific entry format, each entry starting with `deb ...` What is the format? (**HINT**: there is a **man** page for this).

5. Which mirrors are we using by default for APT? Find another mirror that can be used. Can we use more than one source file? (Try this and run `apt-get update`. Why would we want to use more than one source?)

6. What are the "components" at the end of each entry? What is contained within *main*, *contrib*, and *non-free*?

7. What is the difference between the commands `apt-get upgrade` and `apt-get dist-upgrade`?

8. If we wanted to perform a `dist-upgrade`, what are the steps we need to take for doing so?

Besides the command line tools that APT has, there are also several front-end managers that are available. `aptitude`, for example, is a very useful tool for managing dependencies and navigating many of the package trees available. As we continue in the course, at some times using `apt-get` may prove to be unwieldy; run `aptitude` and explore some of the different options and features available.