Guía de referencia Debian

Osamu Aoki <debian@aokiconsulting.com>
Editor: David Sewell <dsewell@virginia.edu>
'Autores' en la página 149

CVS, Sun, 13 Oct 2002 22:40:19 -0600

Resumen

Esta Guía de referencia Debian (`http://qref.sourceforge.net/`) abarca diversos aspectos de la administración del sistema mediante ejemplos que utilizan comandos de la shell. Se brindan tutoriales, trucos e información sobre diversos temas: conceptos básicos del sistema Debian, consejos para la instalación del sistema, administración de paquetes Debian, el kernel de Linux en Debian, puesta a punto del sistema, creación de una puerta de enlace (gateway), editores de texto, CVS, programación y GnuPG para usuarios que no son desarrolladores. Si necesita ayuda para un mantenimiento de emergercia del sistema, diríjase inmediatamente a 'Comandos de supervivencia Debian' en la página 54. La última versión oficial se encuentra en `http://www.debian.org/doc/manuals/debian-reference/` y la última versión en desarrollo en `http://qref.sourceforge.net/Debian/`. El proyecto se encuentra en `http://qref.sourceforge.net/`. Los archivos de configuración de ejemplo se pueden obtener aquí examples (`examples/`).

Nota de Copyright

Índice general

Capítulo 1

Preface

Este documento ha sido empezado como una referencia “rapida”, la información incluida debe ser tratada como complemento ó enlace a las referencias autoritarias listadas en ‘Referencias’ en la página 143.

1.1. Convenciones del documento

El documento “Guía de referencia Debian” proporciona información a través de comandos BASH simples. Aqui siguen las convenciones utilizadas:

```
# comando con cuenta root
$ comando con cuenta "user"
... descripción de la acción.
```

Para BASH, lea `man bash` y Bash-Prog-Intro-HOWTO (LDP) como información para comenzar.

Las siguientes abreviaciones són utilizadas:

- LDP: Linux Documentation Project `http://www.tldp.org/`
- DDP: Debian Documentation Project `http://www.debian.org/doc/`

Ejemplos de scripts están disponibles aquí en el web. Los ficheros escondidos con precediente . sont convertidos con precediente _ (`examples/`).

1.2. Instalación basica

Sí el sistema ha sido instalado con la opción “simple”, asegurarse de seleccionar la opción “New user documentation”. Si no, lanzar los siguientes comandos:

```
# dselect update
# tasksel
... seleccionar la opción "New user documentation" y otras
```

Adicionalmente a esto, instalar los paquetes siguientes:

```
# apt-get install debian-policy developers-reference maint-guide \
       packaging-manual doc-debian doc-linux-text info \
       man-db manpages manpages-dev less mc # para versión Potato

# apt-get install debian-policy developers-reference maint-guide \
       doc-debian doc-linux-text apt-howto info \
       man-db manpages manpages-dev less mc # para versión Woody
```

1.3. Bases de las distribuciones Debian

Debian existe en 3 “versiones” publicas:

- stable: (versión estable) Buena para servidores de producción. Aburrida para utlizar en una workstation (computador de trabajo). ‘The `stable` distribution’ en la página 6
- testing: (versión de pruebas) Buena para probar en una workstation (WS). ‘The `testing` distribution’ en la página 7
- unstable: (versión inestable) Nunca debe fiarse de esta versión. ‘The `unstable` distribution’ en la página 7

Lea a lo menos la lista de correo de desarrollo `debian-devel-announce@lists.debian.org` para información actualiza sobre la evolución de Debian. En marzo 2002, todo esto corresponde a Potato (qualidad de producción), Woody (beta-test, ahora muy estable), y Sid (alpha-test). Cuando los paquetes inestables no tienen bugs de tipo “Release Critical” (RC) desde más o menos la 1era semana, entonces son automaticamente promovidos hacia “testing” (Woody). Vea ‘The Debian archives’ en la página 5.

Teoricamente, para conseguir hacer funcionar las ultimas versiones hay dos posibilidades.

‘Instalar el sistema con APT’ en la página 54 (principalmente para workstations)

‘Port a package to the “stable” system’ en la página 61 (principalmente para servidores)

Después de explicar los fundamentos de las distribuciones Debian en ‘Debian fundamentals’ en la página 5, presentaré alguna información para que vivan felices con las ultimas versiones Debian, conocer las ventajas de las versiones Debian “testing” y “unstable”. Los que no aguanten más siguan ‘Comandos de supervivencia Debian’ en la página 54 inmediatamente. Feliz upgrading (puesta a nivel)! <

Capítulo 2

Debian fundamentals

This chapter provides fundamental information on the Debian system for non-developers. For authoritative information, see:

- Debian Policy Manual
- Debian Packaging Manual (potato)
- Debian Developer's Reference
- Debian New Maintainers' Guide

listed under 'Referencias' en la página 143.

If you are looking for less detailed "how-to" explanations, jump directly to 'Gestión de paquetes Debian' en la página 53 or other relevant chapters.

This chapter consists of documents taken from the "Debian FAQ", greatly reorganized to allow the ordinary Debian system administrator to get started.

2.1. The Debian archives

2.1.1. Directory structures

The software that has been packaged for Debian is available in one of several directory trees on each Debian mirror site (`http://www.debian.org/misc/README.mirrors`) through FTP or HTTP.

The following directories can be found on each Debian mirror site under the `/debian/` directory:

`/dists/`: This directory contains the "distributions", and this used to be the canonical way to access the currently available packages in Debian releases and pre-releases. Some old packages and `Packages.gz` files are still in here.

`/pool/`: New physical location for all packages of Debian releases and pre-releases.

`/tools/`: DOS utilities for creating boot disks, partitioning your disk drive, compressing/decompressing files, and booting Linux.

`/doc/`: The basic Debian documentation, such as the FAQ, the bug reporting system instructions, etc.

`/indices/`: The Maintainers file and the override files.

`/project/`: mostly developer-only materials, such as:

> **`project/experimental/`:** This directory contains packages and tools which are still being developed, and are still in the alpha testing stage. Users shouldn't be using packages from here, because they can be dangerous and harmful even for the most experienced.
>
> **`project/orphaned/`:** Packages that have been orphaned by their old maintainers, and withdrawn from the distribution.

2.1.2. Debian distributions

Normally there are three Debian distributions in the `dists` directory. They are named the "stable" distribution, the "testing" distribution, and the "unstable" distribution. Sometimes there is also a "frozen" distribution. Each distribution is defined as a symlink to the actual directory with a codename in the `dists` directory.

2.1.3. The `stable` distribution

Package entries for the `stable` distribution, Debian Woody (3.0r0), are recorded into the `stable` (symlink to `Woody`) directory:

- `stable/main/`: This directory contains the packages which formally constitute the most recent release of the Debian system.

  These packages all comply with the Debian Free Software Guidelines (http://www.debian.org/social_contract#guidelines) (also available as `/usr/share/doc/debian/social-contrac` installed by `debian-doc`), and are all freely usable and distributable.

- `stable/non-free/`: This directory contains packages distribution of which is restricted in a way that requires that distributors take careful account of the specified copyright requirements.

  For example, some packages have licenses which prohibit commercial distribution. Others can be redistributed but are in fact shareware and not free software. The licenses of each of these packages must be studied, and possibly negotiated, before the packages are included in any redistribution (e.g., in a CD-ROM).

- `stable/contrib/`: This directory contains packages which are DFSG-free and **freely distributable** themselves, but somehow depend on a package that is **not** freely distributable and thus available only in the non-free section.

Now in adition to above locations, new physical packages are located under the `pool` directory ('The `pool` directory' en la página 9).

The current status of `stable` distribution bugs is reported on the Stable Problems (http://ftp-master.debian.org/testing/stable_probs.html) Web page.

2.1.4. The `testing` distribution

Package entries for the `testing` distribution, Debian Sarge, are recorded into the `testing` (symlink to `Sarge`) directory after they have undergone some degree of testing in `unstable`. Now in adition to above locations, new physical packages are located under the `pool` directory ('The `pool` directory' en la página 9). There are also `main`, `contrib` and `non-free` subdirectories in `testing`, which serve the same functions as in `stable`.

These packages must be in sync on all architectures where they have been built and mustn't have dependencies that make them uninstallable; they also have to have fewer release-critical bugs than the versions currently in `unstable`. This way, we hope that `testing` is always close to being a release candidate. More details of the testing mechanism are at http://ftp-master.debian.org/testing/.

The latest status of the `testing` distribution is reported at these sites:

- update excuses (http://ftp-master.debian.org/testing/update_excuses.html)
- testing problems (http://ftp-master.debian.org/testing/testing_probs.html)
- release-critical bugs (http://bugs.debian.org/release-critical/)
- base system bugs (http://base.debian.net/)
- bugs in standard and task packages (http://standard.debian.net/)
- other bugs and bug squashing party notes (http://bugs.debian.net/)

2.1.5. The `unstable` distribution

Package entries for the `unstable` distribution, `sid`, are recorded into the `unstable` (symlink to `sid`) directory after they are uploaded to the Debian archive and stay here until they are moved to `testing`. New physical packages are located under the `pool` directory ('The `pool` directory' en la página 9). There are also `main`, `contrib` and `non-free` subdirectories in `unstable`, which serve the same functions as in `stable`.

The `unstable` distribution contains a snapshot of the most current development system. Users are welcome to use and test these packages, but are warned about their state of readiness. The advantage of using the `unstable` distribution is that you are always up-to-date with the latest in the Debian software project—but if it breaks, you get to keep both parts :-)

The current status of `unstable` distribution bugs is reported on the Unstable Problems (`http://ftp-master.debian.org/testing/unstable_probs.html`) Web page.

2.1.6. The `frozen` distribution

When the `testing` distribution is mature enough, it becomes frozen, meaning no new code is accepted anymore, just bugfixes, if necessary. Also, a new testing tree is created in the `dists` directory, assigned a new codename. The frozen distribution passes through a few months of testing, with intermittent updates and deep freezes called “test cycles”. (The recent `woody` release process did not create a symbolic link `frozen`, so `frozen` was not a distribution but just a development stage of the `testing` distribution.)

We keep a record of bugs in the frozen distribution that can delay a package from being released or bugs that can hold back the whole release. Once that bug count lowers to maximum acceptable values, the frozen distribution becomes stable, it is released, and the previous stable distribution becomes obsolete (and moves to the archive).

2.1.7. Debian distribution codenames

Physical directory names in the `dists` directory, such as `Woody` and `Sarge`, are just “codenames”. When a Debian distribution is in the development stage, it has no version number, but a codename instead. The purpose of these codenames is to make the mirroring of the Debian distributions easier (if a real directory like `unstable` suddenly changed its name to `stable`, a lot of stuff would have to be needlessly downloaded again).

Currently, `stable` is a symbolic link to `Woody`, and `testing` is a symbolic link to `Sarge`. This means that `Woody` is the current stable distribution and `Sarge` is the current testing distribution.

`unstable` is a permanent symbolic link to `sid`, as `sid` is always the unstable distribution.

2.1.8. Codenames used in the past

Other codenames that have already been used are: `buzz` for release 1.1, `rex` for release 1.2, `bo` for releases 1.3.x, `hamm` for release 2.0, `slink` for release 2.1, and `potato` for release 2.2.

2.1.9. The source for codenames

So far they have been characters taken from the movie *Toy Story* by Pixar.

- **buzz** (Buzz Lightyear) was the spaceman,
- **rex** was the tyrannosaurus,
- **bo** (Bo Peep) was the girl who took care of the sheep,
- **hamm** was the piggy bank,
- **slink** (Slinky Dog) was the toy dog,
- **sarge** was a leader of the Green Plastic Army Men,
- **potato** was, of course, Mr. Potato Head,
- **woody** was the cowboy.
- **sid** was a boy next door who destroyed toys.

2.1.10. The `pool` directory

Historically, packages were kept in the subdirectory of `dists` corresponding to the distribution that contained them. This turned out to cause various problems, such as large bandwidth consumption on mirrors when major changes were made.

Packages are now kept in a large “pool”, structured according to the name of the source package. To make this manageable, the pool is subdivided by section (`main`, `contrib` and `non-free`) and by the first letter of the source package name. These directories contain several files: the binary packages for each architecture, and the source packages from which the binary packages were generated.

You can find out where each package is placed by executing a command like `apt-cache showsrc` *`mypackagename`* and looking at the “Directory:” line. For example, the `apache` packages are stored in `pool/main/a/apache/`. Since there are so many `lib*` packages, these are treated specially: for instance, `libpaper` packages are stored in `pool/main/libp/libpaper/`.

The `dists` directories are still used for the index files used by programs like `apt`. Also, at the time of writing, older distributions have not been converted to use pools so you’ll see paths containing distributions such as potato or woody in the Filename header field.

Normally, you won’t have to worry about any of this, as new `apt` and probably older `dpkg-ftp` (see ‘Methods for upgrading a Debian system’ en la página 19) will handle it seamlessly. If you want more information, see the Debian Package Pools FAQ (`http://people.debian.org/~joeyh/poolfaq`).

2.1.11. Historical notes about `sid`

When the present-day `sid` did not exist, the Debian archive site organization had one major flaw: there was an assumption that when an architecture was created in the current `unstable`, it would be released when that distribution became the new `stable`. For many architectures that wasn't the case, with the result that those directories had to be moved at release time. This was impractical because the move would chew up lots of bandwidth.

The archive administrators worked around this problem for several years by placing binaries for unreleased architectures in a special directory called `sid`. For those architectures not yet released, the first time they were released there was a link from the current `stable` to `sid`, and from then on they were created inside the `unstable` tree as usual. This layout was somewhat confusing to users.

With the advent of package pools (see 'The `pool` directory' en la página anterior) during the `woody` distribution development, binary packages began to be stored in a canonical location in the pool, regardless of the distribution, so releasing a distribution no longer causes large bandwidth consumption on the mirrors (there is, however, a lot of gradual bandwidth consumption throughout the development process).

2.1.12. Uploaded packages in `incoming`

Uploaded packages are first located at `http://incoming.debian.org/` after being checked to insure that they really come from a Debian developer and DELAYED for NMU. Once a day, they are moved out of `incoming` to `unstable`.

In an emergency, you may want to install packages from `incoming` before they reach `unstable`.

2.1.13. Retrieve an older package

While the most recent Debian distributions are kept under the `debian` directory on each Debian mirror site (`http://www.debian.org/misc/README.mirrors`), archives for older Debian distributions such as `slink` are kept on `http://archive.debian.org/` or under the `debian-archive` directory on each Debian mirror site.

Older `testing` and `unstable` packages can be located at `http://snapshot.debian.net/`.

2.1.14. Architecture sections

Within each of the major directory trees (`dists/stable/main`,`dists/stable/contrib`,`dists/stable/non-free`, `dists/unstable/main/`, etc.), the binary package entries reside in subdirectories whose names indicate the chip architecture for which they were compiled.

- `binary-all/`, for packages which are architecture-independent. These include, for example, Perl scripts, or pure documentation.

- `binary-`*`platform`*`/`, for packages which execute on a particular binary platform.

Please note that the actual binary packages for `testing` and `unstable` no longer reside in these directories, but in the top-level `pool` directory. The index files (`Packages` and `Packages.gz`) have been kept, though, for backwards compatibility.

For the actual binary architectures supported, see Release Notes for each distribution. They can be located at the Release Notes sites for stable (`http://www.debian.org/releases/stable/releasenotes`) and testing (`http://www.debian.org/releases/testing/releasenotes`).

2.1.15. The source code

Source code is included for everything in the Debian system. Moreover, the license terms of most programs in the system **require** that source code be distributed along with the programs, or that an offer to provide the source code accompany the programs.

Normally the source code is distributed in the `source` directories, which are parallel to all the architecture-specific binary directories, or more recently in the `pool` directory (see 'The `pool` directory' en la página 9). To retrieve the source code without having to be familiar with the structure of the Debian archive, try a command like `apt-get source` *`mypackagename`*.

Some packages, notably `pine`, are only available in source package due to their licensing limitations. (Recently the `pine-tracker` package has been provided to facilitate Pine installation.) The procedures described in 'Port a package to the "stable" system' en la página 61 and 'Creación de paquetes Debian' en la página 138 provide ways to build a package manually.

Source code may or may not be available for packages in the "contrib" and "non-free" directories, which are not formally part of the Debian system.

2.2. The Debian package management system

2.2.1. Overview of Debian packages

Packages generally contain all of the files necessary to implement a set of related commands or features. There are two types of Debian packages:

- **Binary packages**, which contain executables, configuration files, man/info pages, copyright information, and other documentation. These packages are distributed in a Debian-specific archive format (see 'Debian package format' en la página siguiente); they are usually distinguished by having a `.deb` file extension. Binary packages can be unpacked using the Debian utility `dpkg`; details are given in its manual page.

- **Source packages**, which consist of a `.dsc` file describing the source package (including the names of the following files), a `.orig.tar.gz` file that contains the original unmodified source in gzip-compressed tar format, and usually a `.diff.gz` file that contains the Debian-specific changes to the original source. The utility `dpkg-source` packs and unpacks Debian source archives; details are provided in its manual page.

Installation of software by the package system uses "dependencies" which are carefully designed by the package maintainers. These dependencies are documented in the `control` file associated with each package. For example, the package containing the GNU C compiler (`gcc`) "depends" on the package `binutils` which includes the linker and assembler. If a user attempts to install `gcc` without having first installed `binutils`, the package management system (dpkg) will send an error message that it also needs `binutils`, and stop installing `gcc`. (However, this facility can be overridden by the insistent user, see `dpkg(8)`.) For additional details, see 'Package dependencies' en la página 15 below.

Debian's packaging tools can be used to:

- manipulate and manage packages or parts of packages,
- aid the user in the splitting of packages that must be transmitted through a limited-size medium such as floppy disks,
- aid developers in the construction of package archives, and
- aid users in the installation of packages which reside on a remote Debian archive site.

2.2.2. Debian package format

A Debian "package", or a Debian archive file, contains the executable files, libraries, and documentation associated with a particular program suite or set of related programs. Normally, a Debian archive file has a filename that ends in `.deb`.

The internals of this Debian binary package format are described in the `deb(5)` manual page. Because this internal format is subject to change (between major releases of Debian), always use `dpkg-deb(8)` for manipulating `.deb` files.

Through at least the `woody` distribution, all Debian archive files have been manipulable by the standard Unix commands `ar` and `tar`, even when dpkg commands are not available.

2.2.3. Naming conventions for Debian package filenames

The Debian package filenames conform to the following convention:

```
foo_VersionNumber-DebianRevisionNumber.deb
```

where *foo* represents the package name. As a check, one can determine the package name associated with a particular Debian archive file (`.deb` file) in one of these ways:

- inspect the “Packages” file in the directory where it was stored at a Debian archive site. This file contains a stanza describing each package; the first field in each stanza is the formal package name.
- use the command `dpkg --info foo_VVV-RRR.deb` (where *VVV* and *RRR* are the version and revision of the package in question, respectively). This displays, among other things, the package name corresponding to the archive file being unpacked.

The *VVV* component is the version number specified by the upstream developer. There are no standards governing version numbers, so they may have formats as different as “19990513” and “1.3.8pre1”.

The *RRR* component is the Debian revision number, and is specified by the Debian developer (or an individual user if he chooses to build the package himself). This number corresponds to the revision level of the Debian package; thus, a new revision level usually signifies changes in the Debian Makefile (`debian/rules`), the Debian control file (`debian/control`), the installation or removal scripts (`debian/p*`), or in the configuration files used with the package.

2.2.4. Preservation of the local configuration

Preservation of user-configurable files is enabled through Debian’s “conffiles” mechanism. User configuration files (usually placed in `/etc`) are specified in the `conffiles` within the Debian package system. The package management system guarantees not to overwrite these files when the package is upgraded.

When it is possible to configure the system without modifying files that belong to various Debian packages, it is usually a good idea not to modify them even if they are “conffiles”. This ensures faster and smoother upgrade operations.

To determine exactly which files are preserved during an upgrade, run:

```
dpkg --status package
```

and look under “Conffiles:”.

Specifics regarding the contents of a Debian `conffiles` file are provided in the Debian Policy Manual, section 11.7 (see ‘Referencias’ en la página 143).

2.2.5. Debian maintenance scripts

Debian maintenance scripts are executable scripts which are automatically run before or after a package is installed. Along with a file named `control`, all of these files are part of the “control” section of a Debian archive file.

The individual files are:

preinst This script executes before its package is unpacked from its Debian archive (`.deb`) file. Many “preinst” scripts stop services for packages which are being upgraded until their installation or upgrade is completed (following the successful execution of the “postinst” script).

postinst This script typically completes any required configuration of a package once it has been unpacked from its Debian archive (`.deb`) file. Often, ’postinst’ scripts ask the user for input, and/or warn the user that if he accepts default values, he should remember to go back and reconfigure that package as the situation warrants. Many “postinst” scripts then execute any commands necessary to start or restart a service once a new package has been installed or upgraded.

prerm This script typically stops any daemons which are associated with a package. It is executed before the removal of files associated with the package.

postrm This script typically modifies links or other files associated with a package, and/or removes files created by it. (Also see ‘Virtual packages’ en la página siguiente.)

Currently all of the control files can be found in the directory `/var/lib/dpkg/info`. The files relevant to package `foo` begin with the name “foo” and have file extensions of “preinst”, “postinst”, etc., as appropriate. The file `foo.list` in that directory lists all of the files that were installed with the package `foo`. (Note that the location of these files is a dpkg internal, and may be subject to change.)

2.2.6. Package priorities

Each Debian package is assigned a **priority** by the distribution maintainers, as an aid to the package management system. The priorities are:

- **Required** packages are necessary for the proper functioning of the system.

  This includes all tools that are necessary to repair system defects. You must not remove these packages or your system may become totally broken and you may probably not even be able to use dpkg to put things back. Systems with only the Required packages are probably unusable, but they do have enough functionality to allow the sysadmin to boot and install more software.

- **Important** packages should be found on any Unix-like system.

  Other packages without which the system will not run well or be usable will be here. This does **not** include Emacs or X11 or TeX or any other large applications. These packages only constitute the bare infrastructure.

- **Standard** packages are standard on any Linux system, including a reasonably small but not too limited character-mode system.

  This is what will install by default if users do not select anything else. It does not include many large applications, but it does include Emacs (this is more a piece of infrastructure than an application) and a reasonable subset of TeX and LaTeX (if this turns out to be possible without X).

- **Optional** packages include all those that you might reasonably want to install even if you are unfamiliar with them, and if you don't have have specialized requirements.

  This includes X11, a full TeX distribution, and lots of applications.

- **Extra**: packages that either conflict with others with higher priorities, are only likely to be useful if you already know what they are, or have specialized requirements that make them unsuitable for "Optional".

2.2.7. Virtual packages

A virtual package is a generic name that applies to any one of a group of packages, all of which provide similar basic functionality. For example, both the `tin` and `trn` programs are news readers, and should therefore satisfy any dependency of a program that required a news reader on a system, in order to work or to be useful. They are therefore both said to provide the "virtual package" called `news-reader`.

Similarly, `exim` and `sendmail` both provide the functionality of a mail transport agent. They are therefore said to provide the virtual package "mail transport agent". If either one is installed, then any program depending on the installation of a `mail-transport-agent` will be satisfied by the existence of this virtual package.

Debian has a mechanism so that, if more than one package which provides the same virtual package is installed on a system, the system administrator can set one as the preferred package. The relevant command is `update-alternatives`, and is described further in 'Alternative commands' en la página 63.

2.2.8. Package dependencies

The Debian package system has a range of package "dependencies" which are designed to indicate (in a single flag) the level at which Program A can operate independently of the existence of Program B on a given system:

- Package A **depends** on Package B if B absolutely must be installed in order to run A. In some cases, A depends not only on B, but on a specific version of B. In this case, the version dependency is usually a lower limit, in the sense that A depends on any version of B more recent than some specified version.
- Package A **recommends** Package B if the package maintainer judges that most users would not want A without also having the functionality provided by B.
- Package A **suggests** Package B if B contains files that are related to (and usually enhance) the functionality of A.
- Package A **conflicts** with Package B when A will not operate if B is installed on the system. Most often, conflicts are cases where A contains files which are an improvement over those in B. "Conflicts" are often combined with "replaces".
- Package A **replaces** Package B when files installed by B are removed and (in some cases) overwritten by files in A.
- Package A **provides** Package B when all of the files and functionality of B are incorporated into A. This mechanism provides a way for users with constrained disk space to get only that part of package A which they really need.

More detailed information on the use of each these terms can be found in the Packaging manual and the Policy manual.

Note that `dselect` has more fine-grained control over packages specified by **recommends** and **suggests** than `apt-get`, which simply pulls all the packages specified by **depends** and leaves all the packages specified by **recommends** and **suggests**. Both programs in modern form use APT as their back end.

2.2.9. The meaning of Pre-depends

"Pre-depends" is a special dependency. In the case of an ordinary package, `dpkg` will unpack its archive file (i.e., its `.deb` file) independently of whether or not the files on which it depends exist on the system. Simplistically, unpacking means that `dpkg` will extract the files from the archive file that were meant to be installed on your file system, and put them in place. If those packages **depend** on the existence of some other packages on your system, `dpkg` will refuse to complete the installation (by executing its "configure" action) until the other packages are installed.

However, for some packages, `dpkg` will refuse even to unpack them until certain dependencies are resolved. Such packages are said to "pre-depend" on the presence of some other package(s). The Debian project provided this mechanism to support the safe upgrading of systems from `a.out` format to `ELF` format, where the **order** in which packages were unpacked was critical. There are other large upgrade situations where this method is useful, e.g. for packages with "required" priority and their libc dependency.

Once again, more detailed information about this can be found in the Packaging manual.

2.2.10. Package status

Package status can be "unknown", "install", "remove", "purge", or "hold". These "want" flags tell what the user wanted to do with a package (as indicated either by the user's actions in the "Select" section of `dselect`, or by the user's direct invocations of `dpkg`).

Their meanings are:

- unknown - the user has never indicated whether he wants the package.
- install - the user wants the package installed or upgraded.
- remove - the user wants the package removed, but does not want to remove any existing configuration files.
- purge - the user wants the package to be removed completely, including its configuration files.
- hold - the user wants this package not to be processed, i.e., he wants to keep the current version with the current status whatever that is.

2.2.11. Holding back packages from an upgrade

There are two mechanisms of holding back packages from the upgrade, through `dpkg`, or, in woody, through APT.

With `dpkg`, first export the list of package selections:

```
dpkg --get-selections \* > selections.txt
```

Then edit the resulting file *`selections.txt`*, changing the line containing the package you wish to hold, e.g. `libc6`, from this:

```
libc6                         install
```

to this:

```
libc6                         hold
```

Save the file, and reload it into `dpkg` database with:

```
dpkg --set-selections < selections.txt
```

Or, if you know the package name to hold, simply run:

```
echo libc6 hold | dpkg --set-selections
```

This process holds packages at the install process of each package file.

The same effect can be obtained through `dselect`. Simply enter the [S]elect screen, find the package you wish to hold in its present state, and press the '=' key (or 'H'). The changes will take effect immediately after you exit the [S]elect screen.

The APT system in the woody distribution has a new alternative mechanism for holding packages during the archive retrieval process using `Pin-Priority`. See the manual page `apt_preferences(5)`. See `http://www.debian.org/doc/manuals/apt-howto/` or `apt-howto` package.

2.2.12. Source packages

Source packages are distributed in a directory called `source`, and you can either download them manually, or use

```
apt-get source foo
```

to fetch them (see the `apt-get(8)` manual page on how to set up APT for doing that).

2.2.13. Building binary packages from a source package

For a package *foo*, you will need all of *foo*`_*.dsc`, *foo*`_*.tar.gz` and *foo*`_*.`*diff*`.gz` to compile the source (note: there is no `.diff.gz` for a Debian native package).

Once you have them, if you have the `dpkg-dev` package installed, the command

```
$ dpkg-source -x foo_version-revision.dsc
```

will extract the package into a directory called *foo-version*.

If you want just to compile the package, you may cd into the `foo-version` directory and issue the command

```
$ fakeroot debian/rules build
```

to build the program, then

```
$ fakeroot debian/rules binary
```

as root, to build the package, and then

```
# su -c "dpkg -i ../foo_version-revision_arch.deb"
```

to install the newly built package. See 'Port a package to the "stable" system' en la página 61.

2.2.14. Creating new Debian packages

For a more detailed description, read the New Maintainers' Guide, available in the `maint-guide` package, or at `http://www.debian.org/doc/manuals/maint-guide/`.

2.3. Upgrading a Debian system

One of Debian's goals is to provide a consistent upgrade path and a secure upgrade process, and we always do our best to make a new release smoothly upgradable from the previous ones. Packages will alert the user when there are important notices during the upgrade process, and will often provide a solution to a possible problem.

You should also read the Release Notes, the document that describes the details of specific upgrades, shipped on all Debian CDs, and available on the WWW at `http://www.debian.org/releases/stable/releasenotes` or `http://www.debian.org/releases/testing/releasenotes`.

A practical guide to upgrades is provided in 'Gestión de paquetes Debian' en la página 53. This section describes the fundamental details.

2.3.1. Methods for upgrading a Debian system

One could simply execute an anonymous FTP or `wget` call to a Debian archive, peruse the directories until one finds the desired file, fetch it, and finally install it using `dpkg`. Note that `dpkg` will install upgrade files in place, even on a running system. Sometimes, a revised package will require the installation of a newly revised version of another package, in which case the installation will fail until/unless the other package is installed.

Many people find this approach much too time-consuming, since Debian evolves so quickly — typically, a dozen or more new packages are uploaded every week. This number is larger just before a new major release. To deal with this avalanche, many people prefer to use an automated program. Several specialized package management tools are available for this purpose.

2.3.2. Package management tools overview

The Debian package management system has two objectives: the manipulation of the package file itself and the retrieval of package files from the Debian archive. `dpkg` performs the former task, APT and `dselect` the latter.

2.3.3. `dpkg`

This is the main program for manipulating package files; read `dpkg(8)` for a full description.

`dpkg` comes with several primitive supplemental programs.

- dpkg-deb: Manipulate `.deb` files. `dpkg-deb(1)`
- dpkg-ftp: An older package file retrieval command. `dpkg-ftp(1)`
- dpkg-mountable: An older package file retrieval command. `dpkg-mountable(1)`
- dpkg-split: Splits a large package into smaller files. `dpkg-split(1)`

`dpkg-ftp` and `dpkg-mountable` have been superseded by the introduction of the APT system.

2.3.4. APT

APT is an advanced interface to the Debian packaging system. `apt-get`, `apt-cache` and `apt-cdrom` are the command-line tools for handling packages. These also function as the user's "back-end" programs to other tools, such as `dselect` and `aptitude`.

For more information, install the `apt` package and read `apt-get(8)`, `apt-cache(8)`, `apt-cdrom(8)`, `apt.conf(5)`, `sources.list(5)`, `apt_preferences(5)` (woody), and `/usr/share/doc/apt/guide.html/index.html`.

An alternative source of information is the APT HOWTO (`http://www.debian.org/doc/manuals/apt-howto/`). This can be installed by `apt-howto` at `/usr/share/doc/apt-howto/en/apt-howto-en.html/index.html`.

`apt-get upgrade` and `apt-get dist-upgrade` have a tendency to pull in all packages listed under "Suggests:". To avoid this, use `dselect`.

2.3.5. `dselect`

This program is a menu-driven user interface to the Debian package management system. It is particularly useful for first-time installations and large-scale upgrades. See '`dselect` – global configuration' en la página 60.

For more information, install the `install-doc` package and read `/usr/share/doc/install-doc/dselect-beginner.en.html` or dselect Documentation for Beginners (`http://www.debian.org/releases/woody/i386/dselect-beginner`).

2.3.6. Upgrade a running system

The kernel (file system) in Debian systems supports replacing files even while they're being used.

We also provide a program called `start-stop-daemon` which is used to start daemons at boot time or to stop daemons when the kernel runlevel is changed (e.g., from multi-user to single-user or to halt). The same program is used by installation scripts when a new package containing a daemon is installed, to stop running daemons, and restart them as necessary.

Note that the Debian system does not require use of the single-user mode to upgrade a running system.

2.3.7. Downloaded and cached `.deb` archive files

If you have manually downloaded the files to your disk (which is not absolutely necessary, see above for the description of `dpkg-ftp` or APT), then after you have installed the packages, you can remove them from your system.

If APT is used, these files are cached in the `/var/cache/apt/archives/` directory. You may erase them after installation (`apt-get clean`) or copy them to another machine's `/var/cache/apt/archives/` directory to save downloading during subsequent installations.

2.3.8. Record-keeping for upgrades

`dpkg` keeps a record of the packages that have been unpacked, configured, removed, and/or purged, but does not (currently) keep a log of terminal activity that occurred while a package was being so manipulated.

The simplest way to work around this is to run your `dpkg`, `dselect`, `apt-get`, etc., sessions within the `script(1)` program.

2.4. The Debian boot process

2.4.1. The `init` program

Like all Unices, Debian boots up by executing the program `init`. The configuration file for `init` (which is `/etc/inittab`) specifies that the first script to be executed should be `/etc/init.d/rcS`. This script runs all of the scripts in `/etc/rcS.d/` by sourcing or forking subprocess depending on their file extension to perform initialization such as to check and to mount file systems, to load modules, to start the network services, to set the clock, and to perform other initialization. Then, for compatibility, it runs the files (except those with a '.' in the filename) in `/etc/rc.boot/` too. Any scripts in the latter directory are usually reserved for system administrator use, and using them in packages is deprecated. See 'System initialization hints' en la página 87 for more info.

2.4.2. Runlevels

After completing the boot process, `init` executes all start scripts in a directory specified by the default runlevel (this runlevel is given by the entry for `id` in `/etc/inittab`). Like most System V compatible Unices, Linux has 7 runlevels:

- 0 (halt the system),
- 1 (single-user mode),
- 2 through 5 (various multi-user modes), and
- 6 (reboot the system).

Debian systems come with `id=2`, which indicates that the default runlevel will be 2 when the multi-user state is entered, and the scripts in `/etc/rc2.d/` will be run.

In fact, the scripts in any of the directories `/etc/rcN.d/` are just symbolic links back to scripts in `/etc/init.d/`. However, the **names** of the files in each of the `/etc/rcN.d/` directories are selected to indicate the **way** the scripts in `/etc/init.d/` will be run. Specifically, before entering any runlevel, all the scripts beginning with 'K' are run; these scripts kill services. Then all the scripts beginning with 'S' are run; these scripts start services. The two-digit number following the 'K' or 'S' indicates the order in which the script is run. Lower-numbered scripts are executed first.

This approach works because the scripts in `/etc/init.d/` all take an argument which can be either "start", "stop", "reload", "restart" or "force-reload" and will then do the task indicated by the argument. These scripts can be used even after a system has been booted, to control various processes.

For example, with the argument "reload" the command

```
# /etc/init.d/sendmail reload
```

sends the sendmail daemon a signal to reread its configuration file.

2.4.3. Customizing the boot process

Debian does not use a BSD-style `rc.local` directory to customize the boot process; what facilities are provided for doing this?

Suppose a system needs to execute script `foo` on start-up, or on entry to a particular (System V) runlevel. Then the system administrator should:

1. Enter the script `foo` into the directory `/etc/init.d/`.
2. Run the Debian command `update-rc.d` with appropriate arguments, to set up links between the (command-line-specified) directories `rc?.d` and `/etc/init.d/foo`. Here, *?* is a number from 0 through 6 that corresponds to one of the System V runlevels.
3. Reboot the system.

The command `update-rc.d` will set up links between files in the directories `rc?.d` and the script in `/etc/init.d/`. Each link will begin with an 'S' or a 'K', followed by a number, followed by the name of the script. Scripts beginning with 'S' in `/etc/rcN.d/` are executed when runlevel `N` is entered. Scripts beginning with a 'K' are executed when leaving runlevel `N`.

One might, for example, cause the script `foo` to execute at boot-up, by putting it in `/etc/init.d/` and installing the links with `update-rc.d foo defaults 19`. The argument `defaults` refers to the default runlevels, which are 2 through 5. The argument `19` ensures that `foo` is called before any scripts containing numbers 20 or larger.

2.5. Supporting diversity

Debian offers several avenues to accommodate any wishes of the system administrator without breaking the system.

- `dpkg-divert`, see 'dpkg-divert command' en la página 62.
- `equivs`, see '`equivs` package' en la página 63.
- `update-alternative`, see 'Alternative commands' en la página 63.
- `make-kpkg` can accommodate many boot loaders. See `make-kpkg(1)` and 'Debian standard method' en la página 65.

Any files under `/usr/local/` belong to the system administrator and Debian will not touch them. Most (or all) files under `/etc` are `conffiles` and Debian will not overwrite them upon upgrade unless the system administrator requests so explicitly.

2.6. Internationalization

Debian system is internationalized and supports conventions which depend on the language and the cultural rules.

Keyboard Debian is distributed with keymaps for nearly two dozen keyboards, and with utilities (in the `kbd`, package or `console-tools` package with `kbd-compat` package) and to install, view, and modify the tables.

The installation prompts the user to specify the keyboard he will use.

Data The vast majority of Debian software packages support handling non-US-ASCII characters through `locale` technology offered by Glibc.

- Just 8-bit clean: practically all programs
- other Latin languages (e.g. ISO-8859-1 or ISO-8859-2): majority of programs
- multi-byte languages such as Chinese Japanese or Korean: many new applications

Note that Debian does **not** come with all available locales pre-compiled. Check `/usr/lib/locale` to see which locales (besides the default "C") are compiled for your system. See `locale(7),locale-gen(8)` and 'Localization' en la página 109.

Display X can support all fonts. The list includes not only all the 8 bit fonts but also 16 bit fonts such as Chinese Japanese or Korean. See 'CJK and X' en la página 102.

Mamual Currently, support for German-, Spanish-, Finnish-, French-, Hungarian-, Italian-, Japanese-, Korean- and Polish-language manual pages is provided through the `manpages-`*`LANG`* packages (where *LANG* is the two-letter ISO country code).

To access an NLS manual page, the user must set the shell LC_MESSAGES variable to the appropriate string. For example, in the case of the Italian-language manual pages, LC_MESSAGES needs to be set to `it`. The `man` program will then search for Italian manual pages under `/usr/share/man/it/`.

2.7. Debian and the kernel

See 'The Linux kernel under Debian' en la página 65.

2.7.1. Compiling a kernel from non-Debian source

One has to understand the Debian policy with respect to headers.

The Debian C libraries are built with the most recent **stable** releases of the **kernel** headers.

For example, the Debian-1.2 release used version 5.4.13 of the headers. This practice contrasts with the Linux kernel source packages distributed at all Linux FTP archive sites, which use even more recent versions of the headers. The kernel headers distributed with the kernel source are located in `/usr/include/linux/include/`.

If you need to compile a program with kernel headers that are newer than those provided by `libc6-dev`, then you must add `-I/usr/src/linux/include/` to your command line when compiling. This came up at one point, for example, with the packaging of the automounter daemon (`amd`). When new kernels changed some internals dealing with NFS, `amd` needed to know about them. This required the inclusion of the latest kernel headers.

2.7.2. Tools to build custom kernels

Users who wish to (or must) build a custom kernel are encouraged to download the package `kernel-package`. This package contains the script to build the kernel package, and provides the capability to create a Debian kernel-image package just by running the command

```
# make-kpkg kernel_image
```

in the top-level kernel source directory. Help is available by executing the command

```
# make-kpkg --help
```

and through the manual page `make-kpkg(8)` and 'The Linux kernel under Debian' en la página 65.

Users must separately download the source code for the most recent kernel (or the kernel of their choice) from their favorite Linux archive site, unless a kernel-source-*version* package is available (where *version* stands for the kernel version). The Debian `initrd` boot script requires a special kernel patch called `initrd`; see `http://bugs.debian.org/149236`.

Detailed instructions for using the `kernel-package` package are given in the file `/usr/doc/kernel-package/`

2.7.3. Alternative boot loaders

To employ alternative boot loaders such as `grub` or `loadlin`, copy the compiled Linux kernel `bzimage` to other locations (e.g., to `/boot/grub` or to an MS-DOS partition).

2.7.4. Custom boot floppies

The task of making a custom boot floppy is greatly aided by the Debian package `boot-floppies`, normally found in the `admin` section of the Debian FTP archive. Shell scripts in this package produce boot floppies in `syslinux` format. These are MS-DOS formatted floppies whose master boot records have been altered so that they boot Linux directly (or whatever other operating system has been defined in the `syslinux.cfg` file on the floppy). Other scripts in this package produce emergency root disks and can even reproduce the base disks.

You will find more information about this in the `/usr/doc/boot-floppies/README` file after installing the `boot-floppies` package.

2.7.5. Special provisions for dealing with modules

Debian's `modconf` package provides a shell script (`/usr/sbin/modconf`) which can be used to customize the configuration of modules. This script presents a menu-based interface, prompting the user for particulars on the loadable device drivers in his system. The responses are used to customize the file `/etc/modules.conf` (which lists aliases, and other arguments that must be used in conjunction with various modules) through files in `/etc/modutils/`, and `/etc/modules` (which lists the modules that must be loaded at boot time).

Like the (new) Configure.help files that are now available to support the construction of custom kernels, the `modconf` package comes with a series of help files (in `/usr/lib/modules_help/`) which provide detailed information on appropriate arguments for each of the modules. See 'The modularized 2.4 kernel' en la página 67 for examples.

2.7.6. De-installing an old kernel package

The `kernel-image-NNN.prerm` script checks to see whether the kernel you are currently running is the same as the kernel you are trying to de-install. Therefore you can safely remove unwanted kernel image packages using this command:

```
dpkg --purge --force-remove-essential kernel-image-NNN
```

(replace *NNN* with your kernel version and revision number, of course)

Capítulo 3

Debian System installation hints

Official documentation for installing Debian is located at `http://www.debian.org/releases/stable/`, and `http://www.debian.org/releases/stable/installmanual`.

The development versions are located at `http://www.debian.org/releases/testing/`, and `http://www.debian.org/releases/testing/installmanual`.

Although this "Guía de referencia Debian" dates from the days of the potato release, most of its contents are aimed at a Debian Woody (3.0r0) installation.

3.1. General Linux system installation hints

In order to minimize risks associated with "testing" and "unstable" packages, it is a good practice to set up your main Linux system for dual booting along with another small stable Linux system.

3.1.1. Hardware compatibility basics

Linux is compatible with most PC hardware and can be installed to almost any system. For me it was as easy as installing Windoze 95/98/Me. The hardware compatibility list just seems to keep growing.

If you have a laptop PC, check Linux on Laptops (`http://www.linux-laptop.net/`) for installation pointers by brand and model.

My recommendation for desktop PC hardware is "Just be conservative":

- SCSI rather than IDE for work, IDE/ATAPI HD for private use.

- IDE/ATAPI CD-ROM (or CD-RW).
- PCI rather than ISA, especially for the network card (NIC).
- Use a cheap NIC. Tulip for PCI, NE2000 for ISA are good.
- Avoid PCMCIA (notebook) as your first Linux install.
- No USB keyboard, mouse . . . unless you want a challenge.

For a slow machine, yanking out the hard drive and plugging it into another faster machine for installation is a good idea.

3.1.2. Determining a PC's hardware and chip set

During installation, one will be asked to identify hardware or chip set. Sometimes that information may not seem easy to find. Here is my hint:

1. Open a PC box and check inside.
2. Record the numbers on the large chips on the graphics card, network card, chip near serial ports, chip near IDE ports.
3. Record card names printed on the back of the PCI and ISA cards.

3.1.3. Determining a PC's hardware via Debian

The following commands on a Linux system should give some idea of actual hardware and configuration.

```
$ /sbin/lspci -v |pager
$ pager /proc/pci
$ pager /proc/interrupts
$ pager /proc/ioports
```

These commands can be run during the install process from the console screen by pressing ALT-F2.

3.1.4. Determining a PC's hardware via other OSs

Hardware information can also be obtained from other OSs.

Install another commercial Linux distribution. Hardware detection on those tends to be better than on Debian as of now. This may change as woody evolves.

Install Windows. Hardware configuration can be obtained by right-clicking "My Computer" to get to Properties / Device Manager. Record all resource information such as IRQ, I/O port address, and DMA. Some old ISA cards may need to be configured under DOS and used accordingly.

3.1.5. A Lilo myth

Lilo is limited to 1024 cylinders. —WRONG !

The newer `lilo` used in Debian Potato has lba32 support. If the BIOS of your motherboard is recent enough to support lba32, `lilo` should be able to load beyond the old 1024-cylinder limitation.

Just make sure to add a line reading "lba32" somewhere near the beginning of your `lilo.conf` file if you have kept an old `lilo.conf`.

3.1.6. Choice of boot floppies (potato)

An IDEPCI disk set is usually best if you are installing on a desktop. The IDEPCI kernel on the IDEPCI boot disk enables PCI network cards; thus you can get almost everything off the network. Only 2 floppy disks (boot and root) are needed.

For special systems, create a custom rescue disk: replace the kernel image named "linux" on the Debian rescue disk by overwriting it with another compressed kernel image compiled offsite for the machine. Details are documented in `readme.txt` on the rescue disk. The rescue floppy uses the MS-DOS filesystem, so you can use any system to read and edit it. This should make life easier for people with a special network card, etc.

If you have a PCMCIA network card, configure it in the PCMCIA setup; do not set it up in the standard network setup.

3.1.7. Installation

Follow the official instruction described in `http://www.debian.org/releases/stable/installmanual` or `http://www.debian.org/releases/testing/installmanual` (work in progress).

Quick hints for a potato installation from the author's personal experience are:

1. Prepare rescue/root(/driver) disk set. (IDEPCI for a desktop system)
2. Boot with rescue FD and root FD.
3. Run `fdisk` and `fsck`; mount swap, root, tmp, var, home, usr, etc. (no 2.0 kernel support)
4. Install OS. (For IDEPCI, network installable)
5. If the standard kernel FD is used, install driver disks here.
6. Configure drivers. (No action option, if IDE)
7. Install base system from network or FD/CD. (`base2_2.tgz` for potato, `debootstrap` for woody)
8. Configure base system.
9. Do not install `lilo` as MBR and use multi-boot mbr.
10. Reboot the system. (Do not bother creating a boot FD)
11. MD5 passwords “yes”, shadow passwords “yes”; set up an ordinary user account
12. Install “advanced” (dselect **)
13. Select minimum sets to your taste. Exclude emacs, nvi, tex, telnet, talk(d); include mc, vim, kernel-image-2.2.18pre21 (full kernel if IDE disk is used to install) . . .
14. Install (download all. . .)
15. All configuration questions = “y” (replace current)
16. exim: select 2 for machine behind a firewall, 1 for an Internet machine

For more information on dselect, see ‘`dselect` – global configuration’ en la página 60.

3.1.8. Hosts and IP to use for LAN

Example of LAN configuration (C subnet: 192.168.1.0/24):

```
Internet
   |
   +--- External ISP provides POP service (accessed by fetchmail)
   |
  Access point ISP provides DHCP service and SMTP relay service
   |                     :
```

```
    Cable modem         (Dial-up)
      |                      :
  LAN Gateway machine external port: eth0 (IP given by ISP's DHCP)
   use old notebook PC (IBM Thinkpad, 486 DX2 50 MHz, 20 MB RAM)
   run Linux 2.4 kernel with ext3 filesystem.
   run "ipmasq" package (with stronger patch, NAT and firewall)
   run "dhcp-client" package configured for eth0 (override DNS setting)
   run "dhcp" package configured for eth1
   run "exim" as the smarthost (mode 2)
   run "fetchmail" with a long interval (fall back)
   run "bind" as the cache nameserver for Internet from LAN
              as authoritative nameserver for LAN domain from LAN
   run "ssh" on port 22 and 8080 (connect from anywhere)
   run "squid" as the cache server for the Debian archive (for APT)
  LAN Gateway machine internal port: eth1 (IP = 192.168.1.1, fixed)
                             |
           +--- LAN Switch (10 base T) ---+
           |                              |
  Some fixed IP clients on LAN     Some DHCP clients on LAN
  (IP = 192.168.1.2-127, fixed)    (IP = 192.168.1.128-200, dynamic)
```

See 'Building a gateway with a Debian system' en la página 113 for the details of configuring the LAN gateway server.

3.1.9. User accounts

In order to have a consistent feel across machines, the first few accounts are always the same in my system.

I always create a first user account with a name like "admin" (uid=1000). I forward all root email there. This account is made to be a member of the "staff" group, which is given a good amount of root privilege through the `sudo` command. See 'Añadir una cuenta de usuario' en la página 42 for details.

3.1.10. Hard disk partition and NFS setup

I prefer to use different partitions for different directory trees to limit damage upon system crash. E.g.,

```
  /          == (/ + /boot + /bin + /sbin)
```

```
           == 50MB+
/tmp       == 100MB+
/var       == 100MB+
/home      == 100MB+
/usr       == 700MB+ with X
/usr/local == 100MB
```

The size of the `/usr` directory is very dependent on X-window applications and documentation. `/usr` can be 300MB if one runs a console terminal only, whereas 2GB–3GB is not an unusual size if one has installed many Gnome applications. When `/usr` grows too big, moving out `/usr/share/` to a different partition is the most effective cure. With the new large pre-packaged Linux 2.4 kernels, `/` may need more than 200MB.

For example, the current status of my Internet gateway machine is as follows (output of the `df -h` command):

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda3             300M  106M  179M  38% /
/dev/hda7             100M   12M   82M  13% /home
/dev/hda8             596M   53M  513M  10% /var
/dev/hda6             100M  834k   94M   1% /var/lib/cvs
/dev/hda9             596M  222M  343M  40% /usr
/dev/hda10            596M  130M  436M  23% /var/cache/apt/archives
/dev/hda11            1.5G  204M  1.2G  14% /var/spool/squid
```

(The large area reserved for `/var/spool/squid` is for a proxy cache for package downloading.)

Following is `fdisk -l` output to provide an idea of partition structure:

```
# fdisk -l /dev/hda # comment

/dev/hda1            1        41    309928+   6  FAT16 # DOS
/dev/hda2           42        84    325080   83  Linux # (not used)
/dev/hda3   *       85       126    317520   83  Linux # Main
/dev/hda4          127       629   3802680    5  Extended
/dev/hda5          127       143    128488+  82  Linux swap
/dev/hda6          144       157    105808+  83  Linux
/dev/hda7          158       171    105808+  83  Linux
/dev/hda8          172       253    619888+  83  Linux
/dev/hda9          254       335    619888+  83  Linux
/dev/hda10         336       417    619888+  83  Linux
/dev/hda11         418       629   1602688+  83  Linux
```

A few unused partitions exist. These are for installing a second Linux distribution or as expansion space for growing directory trees.

If you have more than 2 hard disks, make a swap partition for each drive to gain maximum performance.

Mounting the above filesystems properly is accomplished with the following `/etc/fstab`:

```
# /etc/fstab: static file system information.
#
# file system mount point type options        dump pass
/dev/hda3 / ext2 defaults,errors=remount-ro 0 1
/dev/hda5 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/fd0 /floppy auto defaults,user,noauto 0 0
/dev/cdrom /cdrom iso9660 defaults,ro,user,noauto 0 0
#
# keep partition separate
/dev/hda7 /home ext2 rw 0 2
/dev/hda8 /var ext2 rw 0 2
/dev/hda6 /var/lib/cvs ext2 rw 0 2
/dev/hda9 /usr ext2 rw 0 2
/dev/hda10 /var/cache/apt/archives ext2 rw 0 2

# very big partition for proxy cache
/dev/hda11 /var/spool/squid ext2 rw 0 2

# backup bootable DOS
/dev/hda1 /mnt/dos vfat rw,noauto 0 0
# backup bootable Linux system (not done)
/dev/hda2 /mnt/linux ext2 rw,noauto 0 0
#
# nfs mounts
mickey:/ /mnt/mickey nfs ro,noauto,intr 0 0
goofy:/ /mnt/goofy nfs ro,noauto,intr 0 0
# minnie:/ /mnt/minnie smbfs ro,soft,intr,credentials={filename} 0 2
```

Here I use `noauto,intr` combined with the default `hard` option for nfs. This way, it is possible to recover from a hung process due to a dead connection using Control-C.

For a Windows machine connected with Samba (smbfs), `rw,auto,soft,intr` may be good idea. (FIXME)

Check-out also autofs (FIXME)

For a floppy drive, using `noauto,rw,sync,user,exec` instead prevents file corruption after accidental disk eject before unmount, but this slows the write process.

The external Linux NFS server (goofy) resides behind a firewall (gateway). I have a very relaxed security policy on my LAN since I am the only user. To enable NFS access, the NFS server side needs to add `/etc/exports` as follows:

```
# /etc/exports: the access control list for file systems which may be exported
#               to NFS clients.  See exports(5).
/       (rw,no_root_squash)
```

This is needed to activate the NFS server in addition to installing and activating the NFS server and client.

For simplicity, I usually create a single partition of 2GB for an experimental and/or secondary lazy Linux install. I optionally share swap and `/tmp` partitions for these installs. A multi-partition scheme is too involved for these usages. If only a simple console system is needed, 500MB may be more than sufficient.

3.1.11. DRAM memory guidelines

Following are rough guidelines for DRAM.

```
   4 MB:  Bare minimum for Linux kernel to function.
  16 MB:  Minimum for reasonable console system.
  32 MB:  Minimum for simple X system.
  64 MB:  Minimum for X system with GNOME/KDE.
 128 MB:  Comfortable for X system with GNOME/KDE.
256+MB:  Why not if you can afford it?  DRAM is cheap.
```

Using the boot option `mem=4m` (or lilo `append="mem=4m"`) will show how the system would perform with 4MB of memory installed. A lilo boot parameter is needed for a system containing more than 64MB memory with an old BIOS.

3.1.12. Swap space

I use the following guidelines:

```
Total swap = min(1x-2x installed RAM, 128 MB - 1 GB)
Each swap  < 128 MB
Keep them on different drives.
Use a central portion of the hard disk when possible.
```

Even if you never need it, some swap space (128MB) is desirable so the system will slow down before it crashes hard with a program which leaks memory.

3.2. Bash configuration

I modify shell start-up scripts to my taste across the system:

```
/etc/bash.bashrc        Replace with private one
/etc/profile            Keep distribution copy ( \w -> \W)
/etc/skel/.bashrc       Replace with private copy
/etc/skel/.profile      Replace with private copy
/etc/skel/.bash_profile Replace with private copy
~/.bashrc               Replace with private copy for all accounts
~/.profile              Replace with private copy for all accounts
~/.bash_profile         Replace with private copy for all accounts
```

See details in my example scripts (`examples/`). I like a transparent system, so I set `umask` to 002 or 022.

`PATH` is set by the following configuration files in this order:

```
/etc/login.defs  - before the shell sets PATH
/etc/profile      (may call /etc/bash.bashrc)
~/.bash_profile   (may call ~/.bashrc)
```

3.3. Mouse configuration

In the case of a PS/2-connector mouse on an ATX motherboard, the signal flow should be:

```
mouse -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X
```

This allows the keyboard and mouse to be unplugged and reinitialized by restarting gpm upon reconnect. X will stay alive!

For a Logitech 3-button PS2 mouse, configuration combinations should be:

```
/etc/gpm.conf                  /etc/X11/X86Config or X86Config4
==============================================================
device=/dev/psaux              Section "Pointer"
responsiveness=                  Protocol    "IntelliMouse"
repeat_type=ms3                  Device      "/dev/gpmdata"
type=ps2auto   (woody)
append=""
---------------------------------------------------------------
device=/dev/psaux              Section "Pointer"
responsiveness=                  Protocol    "IntelliMouse"
repeat_type=raw                  Device      "/dev/gpmdata"
type=ps2auto  (woody)
append=""
```

If a normal 2-button PS2 mouse is used, set the X protocol to `Microsoft` and enable `Emulate3Buttons`. For a scroll mouse, you can adjust X to the real protocol, such as `IMPS/2`. Create a softlink `/dev/gpmdata` –> `/dev/mouse` to make some configuration utilities happy. See my example scripts for details (`examples/`).

For some recent thin Toshiba notebook PCs: Activate `gpm` before PCMCIA in the System-V init script. This keeps `gpm` from crashing. Weird but true.

3.4. NFS configuration

Set up NFS by setting `/etc/exports`.

```
# echo "/ *.domainname-for-lan-hosts(rw,no_root_squash,nohide)" \
        >> /etc/exports
```

See my example scripts for details (`examples/`).

3.5. Samba configuration

Setting up Samba with “share” mode is much easier since this creates WfW-type share drives. But it is preferable to set it up with “user” mode.

Samba can be configured by `sambaconfig` or `vi`:

```
$su -c "sambaconfig"
$su -c "vi /etc/samba/smb.conf"
```

See my example scripts for details (`examples/`).

Adding a new user to the smbpasswd file can be done via `smbpasswd`:

```
$su -c "smbpasswd -a username"
```

Make sure to use encrypted passwords for best compatibility.

Set `os level` according to the following system equivalences (the larger the number, the higher the priority as server):

```
0:      Samba with a loose attitude (will never become a master browser)
1:      Wfw 3.1, Win95, Win98, Win/me?
16:     Win NT WS 3.51
17:     Win NT WS 4.0
32:     Win NT SVR 3.51
33:     Win NT SVR 4.0
255:    Samba with mighty power
```

Make sure that users are members of the group owning the directory that gives shared access and that the directory path has its execution bit set to access.

3.6. Printer configuration

You may install LPRng (or GNUlpr) instead of standard `lpr`.

```
# apt-get install lprng enscript gs
```

As of 3/2001 Debian potato2.2r2, `printtool` and `lprngtool` are not in potato. Grab one of these from `woody/binary-all/admin` and `dpkg -i` to install.

If Windoze machines use LPRng through Samba, they can access it without GS-filtering through a native Windows printer driver or with GS-filtering through PostScript printer drivers such as Apple LaserWriter.

3.7. Other host configuration hints

3.7.1. Install a few more packages and do the basics

See '`dselect` – global configuration' en la página 60. I usually dump (type '_'): TEX, nvi, ae, lynx, and add (type '+'): vim, ssh, lynx-ssl, mc.

Edit `/etc/inittab` to have CTRL-DEL-ALT=halt for easy shutdown.

3.7.2. Modules

Done while configuring drivers during installation. List module names in `/etc/modules`. I also use `lsmod` and `depmod` to control them manually.

Also add a few lines in `/etc/modules` to handle ip-masquerading (ftp, etc.) for 2.4 kernels. See my example scripts for details (`examples/`).

3.7.3. CD-RW basic setup

Edit the following files:

```
/etc/lilo.conf   (add append="hdc=ide-scsi ignore=hdc",
                  run lilo to activate)
/dev/cdrom       (softlink # cd /dev; ln -sf scd0 cdrom)
/etc/modules     (add "ide-scsi" and "sg".  If needed "sr" after this.)
```

See 'CD-writer' en la página 91 for details.

3.7.4. Large memory and auto power-off

Edit `/etc/lilo.conf` as follows to set boot prompt parameters for large memory (for 2.2 kernels) and auto power-off (for apm):

```
append="mem=128M apm=on apm=power-off"
```

Run `lilo` to install these settings. `apm=power-off` is needed for an SMP-kernel. The same can be done directly by entering options at the boot prompt. See 'Otros trucos con el indicador de arranque' en la página 73.

If apm is compiled as a module, as in Debian default 2.4 kernels, run `# insmod apm power_off=1` after boot or set `/etc/modules` by:

```
# echo "apm power_off=1" >>/etc/modules
```

Alternatively, compiling ACPI support achieves the same goal with newer kernels and seems to be more SMP-friendly (this requires a newer motherboard). The 2.4 kernel on newer motherboards should detect large memory correctly.

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

and add the following lines in `/etc/modules` in this order:

```
ospm_busmgr
ospm_system
```

Or recompile the kernel with all of the kernel options above set to “y”. In any case, none of the boot prompt parameters are needed with ACPI.

3.7.5. Other configuration files to tweak in `/etc`

You may want to add an `/etc/cron.deny` file, missing from the standard Debian install (you can copy `/etc/at.deny`).

Capítulo 4

Tutoriales de Debian

Esta sección proporciona una orientación al mundo Linux para aquellos que son realmente novatos. Si ya hace tiempo que usa Linux, considérelo como un repaso.

4.1. Fuentes de información

Visite el Proyecto de Documentación Debian (DDP) (`http://www.debian.org/doc/`) que tiene las referencias más importantes sobre Debian. Muchos de estos documentos se encuentran instalados en `/usr/share/doc/`. También consulte `/usr/share/doc-base/` que proporcionan enlaces a los documentos del sistema. Añada `export CDPATH=.:/usr/share/doc:/usr/src/local` al `~/.bash_profile` para un acceso más cómodo a los directorios de documentación.

El Proyecto de documentación Linux (LDP) (`http://www.tldp.org/`) tiene las referencias más importantes sobre Linux en general. Los contenidos del LDP usualmente están instalados en `/usr/share/doc/HOWTO/`.

Navegue por los documentos en forma local y en ftps remotos con la tecla `F9` del Midnight Commander (véase 'Midnight Commander (MC)' en la página 44).

4.2. La consola Linux

4.2.1. Entrando al sistema

En un sistema Linux existen 6 seudo-terminales independientes. Se puede pasar de una a otra presionando simultáneamente la tecla `Alt-izq` con la tecla `F1` - `F6`. Cada seudo-terminal permite el ingreso al sistema en forma independiente a diferentes cuentas. El sistema multi-usuario es una gran característca de los sistemas Unix y resulta muy adictiva.

En Unix se considera un buen hábito acceder al sistema como usuario normal para realizar la mayoría de las tareas. Debo reconocer que, por pereza, utilizo la cuenta del superusuario (root) más de lo necesario.

Generalmente utilizo una cuenta normal con los comandos `sudo`, `super` o `su -c` para obtener accesos limitados de root.

4.2.2. Añadir una cuenta de usuario

Tras la instalación del sistema, añada un cuenta de usuario normal. Si el nombre de usuario es "penguin",

```
# adduser penguin
```

creará dicha cuenta.

Utilice el comando `vigr` para editar el `/etc/group` de la siguiente manera:

```
src:x:40:admin, debian, ...
staff:x:50:admin
...
```

Consulte `adduser`, `addgroup`, `vipw`, `vipw -s`, `vigr` y `vigr -s` para configurar correctamente a usuarios y grupos.

4.2.3. Cómo apagar el sistema

Al igual que muchos sistemas operativos modernos donde los archivos son almacenados en memoria, Linux necesita apagarse correctamentes antes de poder cortar el suministro eléctrico sin ningún peligro. Veamos el comando para el modo multiusuario:

```
# shutdown -h now
```

Y el siguiente para el modo monousuario:

```
# poweroff -i -f
```

Espere a que aparezca el mensaje "System halted" y a continuación apague la máquina. Si apm está habilitado tanto en la BIOS como en Linux, el sistema se apagará por sí solo. Véase 'Large memory and auto power-off' en la página 38 para más detalles.

4.2.4. Edición en línea de comandos

El intérprete de comandos por defecto, `bash` tiene la capacidad de navegar por el histórico de comandos. Simplemente utilize la tecla flecha-arriba para entrar en él. Otras combinaciones de teclas importantes para tener en cuenta:

```
Ctrl-C:                       Finaliza un programa
Ctrl-D:                       Finaliza una entrada
Ctrl-S:                       Detiene la salida por pantalla
Ctrl-Q:                       Reactiva la salida por pantalla
Ctrl-Alt-Supr:                Reinicia/apaga el sistema (véase /etc/inittab)
Lt-pulsar-y-arrastrar-ratón: Selecciona y copia al portapapeles (gpm)
Ctrl-pulsar-ratón:            Pega el contenido del portapapeles en el lugar
                              donde se encuentra el cursor (gpm)
```

En una consola Linux tipo únicamente funcionan correctamente las teclas `Ctrl` y `Alt` situadas a la izquierda del teclado.

4.2.5. Comandos básicos que se deben tener presente

Los siguientes son comandos básicos de Unix:

```
ls, ls -al, ls -d, pwd, cd, cd ~user, cd -,
cat /etc/passwd, less, bg, fg, kill, killall,
uname -a, type nombre_comando, sync, netstat,
ping, traceroute, top, vi, ps aux, tar, zcat,
grep, ifconfig, ...
```

Averigüe su significado escribiendo el comando, usando `man` o `info` seguidos por el nombre del comando. En Linux, muchos comandos muestran una breve ayuda informativa si se los invoca de una de las siguientes maneras:

```
$ nombre_comando --help
$ nombre_comando -h
```

`whatis` *`nombre_comando`* brinda una resumen de cualquier comando del sistema que tenga una entrada en el manual.

4.2.6. Sistema X Window

Para iniciar el Sistema X Window desde la consola:

```
# exec startx
```

Pulsando con el botón derecho del ratón sobre la ventana raíz aparecerá un menú desplegable.

4.2.7. Combinaciones de teclas habituales

Algunas combinaciones de teclas importantes para tener en cuenta cuando se está en la consola de Linux:

```
Alt-F1/F6:           Para cambiar a otra seudo-terminal
Ctrl-Alt-F1/ F6:     Para cambiar a otra seudo-terminal
                     (desde X-Window, DOSEMU, etc.)
Alt-F7:              Para volver a  X-Window
Ctrl-Alt-menos:      Cambiar la resolución de la pantalla en X-Window
Ctrl-Alt-más:        Cambiar la resolución de la pantalla en X-Window en
                     sentido opuesto
Ctrl-Alt-Retroceso:  Finalizar X-Window
Alt-X, Alt-C, Alt-V: En algunos programas como en 'Netscape Composer',
      las combinaciones de teclas usadas con Ctrl
      utilizados en Windows/Mac para Cortar, Copiar y
      Pegar se reemplazan por las mismas combinaciones
      pero usando la tecla Alt.
```

4.3. Midnight Commander (MC)

Midnight Commander (MC) es la “navaja suiza” GNU para la consola de Linux y otros entornos de terminales.

4.3.1. Instalar MC

```
# apt-get install mc
```

A continuación añada la siguiente función al `~/.bashrc` (o al `/etc/bash.bashrc` que es llamado desde el `.bashrc`).

```
mc ()
{
  mkdir -p ~/.mc/tmp 2> /dev/null
  chmod 700 ~/.mc/tmp
  MC=~/.mc/tmp/mc-$$
  /usr/bin/mc -P "$@" > "$MC"
  cd "$(cat $MC)"
  rm -f "$MC"
  unset MC;
}
```

Al salir, esto habilita al MC a cambiar al directorio de trabajo.

Si utiliza una terminal como `kon` o `Kterm` para el idioma japonés que emplea determinados caracteres gráficos el agregar la opción `-a` a la línea de comando del MC puede ayudar a evitar problemas.

4.3.2. Iniciar MC

```
$ mc
```

MC se encarga de todas las operaciones con archivos mediante menús requiriendo de un mínimo esfuerzo por parte del usuario.

4.3.3. Administrador de archivos

Por defecto, se tienen dos paneles con el listado de archivos de un directorio. Otro modo útil consiste en configurar la ventana derecha para que muestre toda la información referente a los archivos: permisos, tamaño, etc. Los siguientes son algunas teclas esenciales. Con el demonio `gpm` ejecutándose es posible también usar el ratón. Asegúrese de presionar la tecla Mayús para cortar y pegar en el MC.

- `F1`: Menú de ayuda
- `F3`: Visor de archivos interno
- `F4`: Editor interno
- `F9`: Activar menú desplegable
- `F10`: Salir del Midnight Commander

- `Tab`: Moverse entre las dos ventanas
- `Insert`: Marcar archivo para operaciones con múltiples archivos
- `Supr`: Borrar archivo (sea cuidadoso—configure MC para el modo de borrado seguro)
- Teclas de desplazamiento: No necesita explicación alguna

4.3.4. Trucos para la línea de comandos:

- Cualquier comando `cd` cambiará el directorio mostrado en los paneles.
- `Control-Enter` o `Alt-Enter` copiará el nombre de un archivo en la línea del comandos. Utilice este atajo con los comandos `cp` o `mv`.
- `Alt-Tab` cumple el mismo rol que la tecla TAB en el shell.
- Se puede especificar el directorio inicial para ambas ventanas ingresándolos como argumentos del MC; por ejemplo, `mc /etc /root`.
- `Esc` + *número* == `Fn` (es decir, `Esc` + `1` = `F1`, etc.; `Esc` + `0` = `F10`)
- Tecla `Esc` == tecla `Alt` (= `Meta`, `M-`); es decir, escriba `Esc` + `c` para `Alt-c`

4.3.5. Editor

El editor interno sigue un esquema de cortar-y-pegar muy interesante. Con `F3` se marca el comienzo de una selección, un segundo `F3` marca el fin de la misma y resalta el área elegida. A continuación se puede mover el cursor. Si pulsa `F6` el área seleccionada se ubicará donde está el cursor. Presionando `F5` el área se copiará e insertará en dicha posición. `F2` grabará el archivo. Con `F10` se sale del editor. La mayoría de las teclas de desplazamiento funcionan en forma intuitiva.

Este editor se puede iniciar directamente junto con un archivo:

```
$ mc -e nombre_archivo_a_editar
$ mcedit nombre_archivo_a_editar
```

Si bien no se trata de un editor multiventana se pueden usar múltiples consolas Linux para lograr el mismo efecto. Para copiar entre ventanas, utilice las teclas `Alt-Fn` para alternar entre las consolas virtuales y use "Archivo->Insertar archivo" o "Archivo->Copiar a archivo" para mover una porción de un archivo en otro archivo.

El editor interno se puede reemplazar por cualquier editor externo de su preferencia.

Asimismo, muchos programas usan las variables de entorno `EDITOR` o `VISUAL` para decidir qué editor usar. Si no se siente cómodo con vim, iguale estas variables a `mcedit` añadiendo las siguientes líneas al `~/.bashrc`:

```
...
export EDITOR=mcedit
export VISUAL=mcedit
...
```

En lo posible, es recomendable dejarlas en `vim`. Acostumbrarse con los comandos de vi(m) es proceder de forma correcta pués es un editor de uso habitual en el mundo Linux/Unix.

4.3.6. Visor

Es un visor muy sofisticado. Es una excelente herramienta para buscar palabras en un documento. Siempre lo uso para los archivos del directorio `/usr/share/doc`. Esta es la manera más rápida de navegar por la inmensa cantidad de información de Linux. Este visor se puede arrancar directamente de la siguiente manera:

```
$ mc -v nombre_archivo
```

(Obsérvese que algunos paquetes no respetan esta convención y almacenan sus documentos en `/usr/doc`)

4.3.7. Inicio automático de programas

Pulse `Enter` sobre un archivo y el programa apropiado manejará el contenido del archivo. Esta es una característica muy útil del MC.

```
ejecutable:           Ejecuta el programa
man, archivo html:    Deriva el contenido a un visor
tar, gz, archivo rpm: Permite ver su contenido incluyendo subdirectorios
```

Para que esto funcione, los archivos no deben ser ejecutables. Cambie su estado usando el comando chmod mediante el menú 'Archivo' del MC si fuese necesario.

4.3.8. Sistema de archivos virtuales FTP

Se puede usar el MC para acceder a archivos que se encuentran en Internet via FTP. Vaya al menú presionando `F9,` luego pulse `p` para activar el sistema de archivos virtual FTP. Ingrese la URL de la siguiente manera `nombre_usuario:contraseña@nombre_servidor.nombre_dominio` y se verá al directorio remoto como si fuera local.

4.4. Para saber más

Existen muy buenas referencias de iniciación Unix sobre el tema. Los libros de O'Reilly son, en general, excelentes guías para prácticamente cualquier tema de informática .Recomiendo *Unix Power Tools* que está lleno de información.

Tips-HOWTO (LDP) es otro recurso para tener en cuenta.

Capítulo 5

Transición hacía Woody

5.1. Preparación para la transición

Una actualización hacía la version “testing” via la Red es posible mediante el siguiente método (lo mejor es hacer un script como el siguiente go-woody (`http://qref.sourceforge.net/quick/examples/go-woody`) para lanzar el proceso en un solo comando):

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
# cd /
# apt-setup noprobe
 ... select http or ftp
# cd /etc/apt
# grep -e "^deb " sources.list             >sources.deb
# grep -e "^deb-" sources.list             >sources.src
# sed -e "s/^d/#d/" \
>  /usr/share/doc/apt/examples/sources.list
>  >sources.list
# sed -e "s/stable/testing/"  \
>  sources.deb >>sources.list
# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
> Package: *
> Pin: release a=testing
> Pin-Priority: 700
>
```

```
> Package: *
> Pin: release a=unstable
> Pin-Priority: 70
>
> EOF
# sed -e "s/stable/unstable/" sources.deb \
>  >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
> sed -e "s/^deb-/#deb-/"  >>sources.list
```

Una guía para `/etc/apt/preferences` (consulte `apt_preferences(5)`):

```
track stable:            cambiar  Pin-Priority de testing  a  80
track testing:           dejar como está (instalar unstable por /unstable)
track testing(unstable): cambiar  Pin-Priority de unstable a  600
track unstable(testing): cambiar  Pin-Priority de unstable a  800
```

Esta tabla sirve para decidir el orden de la variable Pin-Priority según el orden de aparición (de arriba a abajo) y esto siguiendo el orden de transición de una versión distribuable a una versión "freeze".

Asegúrese de configurar `apt` para utilizar un proxy, si necesario, configurando la variable `http_proxy` o completando la valor http en el fichero `/etc/apt/apt.conf`.

La procedimiento descrito en esta sección solo actualiza `apt` y los paquetes necesarios para evitar problemas de dependencias.

Since this upgrade method uses `apt-get`, its handling of *recommends* and *suggests* is limited. Read 'Package dependencies' en la página 15 and use `dselect` instead to achieve fine-grained control.

5.2. Actualización a Woody

Después de la preparación precedente utilize los comandos:

```
# apt-get update # siempre lanzar este comando antes del "upgrade"
 ... este comando sirve para actualizar todo el sistema con los paquetes sugerido
# apt-get -u dist-upgrade
 ... para actualizar el sistema y guardar la configuración dselect actual.
# apt-get -u dselect-upgrade       # utiliza la configuración dselect para actual
```

5.3. Configuración de Woody

Esto no sera necesario a medida que Woody se mejore.

Para un sistema recientemente instalado con Woody, edite el fichero `/etc/apt/sources.list`, `/etc/apt/apt.conf`, y `/etc/apt/preferences` para conseguir la misma estructura que la descrita en las secciones precedentes.

5.4. `sources.list` Optimizado

Optimize `sources.list` probando la respuesta (latency) y anchura de banda (bandwidth) de cada sitio. Ejemplo utilizando apt-spy:

```
# apt-get install apt-spy
# cd /etc/apt ; mv sources.list sources.list.org
# apt-spy -d testing -l sources.apt
```

Capítulo 6

Gestión de paquetes Debian

Asegúrese de configurar un proxy HTTP local utilizando `squid` para los paquetes bajados con APT. Esto mejora considerablemente las actualizaciones via la Red, especialment con varios sistemas Debian en la LAN. Este documento está basado en el sistema Woody pero tambíen se aplica a Potato (excepto para `apt_preferences(5)` y asuntos relativos a `/etc/preferences)`).

6.1. Introducción

Si esto es mucho pedir, lea este documento primero y empieze a disfrutar de la potencia de Debian con testing/unstable :-)

6.1.1. Herramientas principales

```
dselect      - Gestion de paquetes via menu (primer nivel)
apt-get      - Instala un paquete (package-archive centric, APT)
dpkg         - Instala un paquete (package-file centric)
aptitude & deity - Futuros remplazos para dselect (GUI APT)
```

Estas herramientas no son todas de mismo nivel. `dselect` se ejecuta por encima de APT (las líneas de comando son: `apt-get`) y `dpkg`. Cuando se utilize `dselect`, asegurarse de actualizar el fichero de estado (utilizando [U]pdate en el menu) antes de la selección si ha instalado paquetes con `apt-get`.

As for package dependencies, `apt-get` automatically pulls in packages with *depends* but leaves packages with *suggests* and *recommends*, while `dselect` offers fine-grained control over choices of these packages. See 'Package dependencies' en la página 15.

6.1.2. Herramientas convenientes

```
apt-cache         - consulta el archivo de paquetes en el "cache" local
dpkg-reconfigure  - lanza el proceso de reconfiguración de un paquete ya instalac
dpkg-source       - gestion de los paquetes de codigo fuente
dpkg-buildpackage - construcción automatizada de paquetes
...
```

6.2. Comandos de supervivencia Debian

Con este conocimiento, se puede vivir eternamente de “upgrades” :-)

Refiérase tambíen a ‘Debian System installation hints’ en la página 27 and ‘Transición hacía Woody’ en la página 49.

6.2.1. Instalar *task* con `tasksel`

`tasksel` es el *Debian Task Installer*, el cúal es el “`simple`” método para instalar el sistema.

Cuando se necesita instalar una función común que requiere varios paquetes esta es la mejor manera. Asegúrese de ejecutar como sigue :

```
# dselect update
# tasksel
```

6.2.2. Instalar el sistema con APT

Se puede instalar selectivamente paquetes desde archivos diferentes utilizando versiones más recientes de `apt-get` (>Woody). Esto activará la actualización selectiva hacia “unstable” y desactualización selectiva hacia “stable” mientras se sigue vigilando “testing”.

```
# apt-cache policy libc6 libc6-dev locales         # verifica el estado
# apt-get install libc6=2.2.4-1 libc6-dev=2.2.4-1 locales=2.2.4-1
# apt-get install libc6/unstable libc6-dev/unstable locales/unstable
# apt-get install -t unstable libc6 libc6-dev locales
# apt-get -u install interesting-new-package remove-package-
# apt-get remove useless-old-package
# apt-get remove --purge really-useless-old-package
```

Para desactualizar todos los paquetes hacia “stable”, modifique `/etc/apt/preferences` como sigue:

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

y lanze `apt-get upgrade`, esto provoca la desactualización por causa de la variable Pin-priority > 1000.

6.2.3. Actualizar con APT

Actualizar el sistema con APT:

```
# apt-get update
... siguiendo con lo siguiente:
# apt-get -u upgrade         # instala según recomendado
# apt-get -u dist-upgrade    # instala lo recomendado y verifica las dependencia
# apt-get -u dselect-upgrade # sigue la selección con dselect
```

Utilize la opción `-s` para simular la actualización sin hacerlo realmente.

`dselect` ofrece un interfaz de tipo menu por encima de APT. `deity` y `aptitude` ofrecerán alternativas a `dselect`.

6.2.4. Verificar los bugs Debian

A menudo, la mayor parte de los problemas ya son conocidos. Comienze primero consultando aquí:

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/<packagename>
```

Buscar con Google (www.google.com) con palabras clave como “site:debian.org”.

Si duda consulte el manual. Modifique `CDPATH` como sigue:

```
export CDPATH=.:/usr/local:/usr/share/doc
```

y ejecute

```
$ cd <packagename>
$ mc <packagename>
```

6.2.5. Solución de problemas de actualización APT.

Problemas de dependencias de paquetes pueden ocurrir cuando se actualiza en unstable/testing. frecuentemente, esto ocurre cuando un paquete debe ser actualizado y una nueva dependencia no es respetada. Estos problemas se solucionan utilizando:

```
# apt-get dist-upgrade
```

Sí esto no funciona, entonces repita lo siguiente hasta que el problema se solucione por si solo:

```
# apt-get upgrade -f         # continua la actualización mismo despues de un erro
... or
# apt-get dist-upgrade -f    # continua dist-upgrade mismo despues de un error.
```

Algún script de actualización corrupto puede causar daños definitivos. Es mejor resolver este tipo de problemas inspectando los ficheros `/var/lib/dpkg/info/nombredelpaquete.`*{post-,pre-}{instal* del paquete corrupto y luego lanzando:

```
# dpkg --configure -a    # configura todos los paquetes parcialmente instalados
```

Si un script se queja de la falta de un fichero de configuración, mire en `/etc` para encontrarlo. Si existe un fichero con la extensión `.new` (o algo similar), cambielo (`mv`) quitando el sufijo.

Pueden aparecer problemas de dependencias cuando se instalan paquetes unstable/testing. Es posible sobrepasar las dependencias.

```
# apt-get install -f package # ignora la dependencias rotas
```

El método alternativo para reparar estas situaciones es utilizar el paquete `equivs`. Vea '`equivs` package' en la página 63.

6.2.6. Reparación utilizando `dpkg`

Una reparación apropiada para un crash `dselect` (APT) es utilizando `dpkg` sin APT:

```
# cd /var/cache/apt/archives
# dpkg -i libc6* libdb2* perl*
# dpkg -i apt* dpkg* debconf*
# dpkg -i *  (si no hay errores)
```

Si falta un paquete, cogerlo por:

```
# mc            # utilize "FTP link" dirigido sobre el servidor FTP Debian
```

Como recientemente, los paquetes actuales en los servidores HTTP/FTP puede ser que no se encuentren en el clasico dosier `/dist` en cambio mire en el nuevo `/pool`.

Luego instale así:

```
# dpkg -i /var/cache/apt/archives/elpaquete.deb
```

Para una dependencia rota, repare o utilize:

```
# dpkg --ignore-depends=paquete1,... -i elpaquete.deb
# dpkg --force-depends -i elpaquete.deb
# dpkg --force-depends --purge nombredelpaquete
```

6.2.7. Instalar un paquete en un sistema no bootable

Boot Linux a partir de un floppy, CD, o otra partición si se trata de un sistema Linux multiboot. Montar la partición sistema no bootable en `/target` y utilize el modo de instalación chroot de `dpkg`.

```
# dpkg --root /target -i elpaquete.deb
```

Luego configure y corriga los problemas.

By the way, if a broken `lilo` is all that prevents booting, you can boot using a standard Debian rescue disk. At boot prompt, assuming the root partition of your Linux installation is in `/dev/hda12` and you want runlevel 3, enter:

```
boot: rescue root=/dev/hda12 3
```

Then you are booted into an almost fully functional system with the kernel on floppy disk. (There may be minor glitches due to lack of kernel features or modules.)

6.3. Debian nirvana commands

Enlightment with these commands will save a person from the eternal karmic struggle of upgrade hell and let him reach Debian nirvana. :-)

6.3.1. Information on a file

To find the package to which a particular file belongs:

```
$ dpkg {-S|--search} pattern # search package from installed filename
$ zgrep -e pattern /local/copy/of/debian/woody/Contents-i386.gz
             # find filename-pattern of files in the debian archive
```

Or use specialized package commands:

```
# apt-get install dlocate
                 # conflicts with slocate (secure version of locate)
$ dlocate filename         # fast alternative to dpkg -L and dpkg -S
...
# apt-get install auto-apt # on-demand package installation tool
# auto-apt update          # create db file for auto-apt
$ auto-apt search pattern
                 # thorough search over the archive of packages
```

6.3.2. Information on a package

Search and display information from package archives. Make sure to point apt to the proper archive(s) by editing `/etc/apt/sources.list`. If you want to see how packages in testing/unstable do against the currently installed one, use `apt-cache policy`—quite nice.

```
# apt-get   check          # update cache and check for broken packages
$ apt-cache search  pattern # search package from text description
$ apt-cache policy  package # package priority/dists information
$ apt-cache show -a package # show description of package in all dists
$ apt-cache showpkg package # package information for debugging
# dpkg  --audit|-C          # search for partially installed packages
$ dpkg {-s|--status} package ... # description of installed package
$ dpkg -l package ...       # status of installed package (1 line each)
$ dpkg -L package ...       # list file names installed by the package
```

You can also find package information in (I use `mc` to browse these):

```
/var/lib/apt/lists/*
/var/lib/dpkg/{available|status}
```

6.3.3. Reconfigure installed packages

Use the following to reconfigure any already-installed package.

```
# dpkg-reconfigure --priority=medium package [...]
# dpkg-reconfigure --all  # reconfigure all packages
```

Do this for `debconf` if you need to change the `debconf` dialog mode permanently.

Some programs come with special configuration scripts.

```
apt-setup     - create /etc/sources.list
install-mbr   - install a Master Boot Record manager
tzconfig      - set the local timezone
gpmconfig     - set gpm mouse daemon
smbconfig     - configure Samba
eximconfig    - configure Exim (MTA)
texconfig     - configure teTeX
apacheconfig  - configure Apache (httpd)
cvsconfig     - configure CVS
sndconfig     - configure sound system
...
update-alternatives - set default command, e.g., vim as vi
update-rc.d         - System-V init script management
update-menus        - Debian menu system
...
```

6.3.4. Remove and purge packages

Remove a package while maintaining its configuration:

```
# apt-get remove package ...
# dpkg  --remove package ...
```

Remove a package and all configuration:

```
# apt-get remove --purge package ...
# dpkg    --purge        package ...
```

6.3.5. Holding older packages

For example, holding of `libc6` and `libc6-dev` for `dselect` and `apt-get -u upgrade package` can be done as follows:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`apt-get -u install package` will not be hindered by this “hold”. To hold a package through forcing automatic downgrade for `apt-get -u upgrade package` or `apt-get -u dist-upgrade`, add the following to `/etc/apt/preferences`:

```
Package: libc6*
Pin: release a=stable
Pin-Priority: 2000
```

The following will list packages on hold:

```
dpkg --get-selections "*"|grep -e "hold$"
```

6.3.6. `dselect` – global configuration

Add a line with “expert” in `/etc/dpkg/dselect.cfg` to reduce noise.

When started, `dselect` automatically selects all “Required”, “Important”, and “Standard” packages. Some packages, such as `teTEX` and `emacs`, are large and may not be needed when you first install a system. So be careful.

`dselect` has a somewhat strange user interface. There are 4 ambiguous commands (Capital means CAPITAL!):

```
Key-stroke  Action
Q           Quit. Confirm current selection and quit anyway.
            (override dependencies)
R           Revert! I did not mean it.
D           Damn it! I do not care what dselect thinks.  Just Do it!
U           Set all to sUggested state
```

With `D` and `Q`, you can select conflicting selections at your own risk. Handle these commands with care. For a slower machine, run `dselect` on another fast machine to find packages and use `apt-get install` to install them. `apt-get dselect-upgrade` best honors `dselect` selection.

6.3.7. Reduce cached package files

Package install with APT leaves cached package files in `/var/cache/apt/archives` and these need to be cleaned.

```
# apt-get autoclean # removes only useless package files
# apt-get clean     # removes all cache package files
```

6.3.8. Record/copy system configuration

To make a local copy of the package selection states:

```
$ dpkg --get-selections "*" >myselections   # or use \*
```

`"*"` makes `myselections` include package entries for “purge” too.

You transfer this file to another computer, and install it there with:

```
# apt-get update
# dpkg --set-selections <myselections
# apt-get -u dselect-upgrade
```

6.3.9. Port a package to the “stable” system

For partial upgrades of the stable system, rebuilding a package within its environment using the source package is desirable. This avoids massive package upgrades due to their dependencies. First, add the following entries to `/etc/apt/sources.list`:

```
deb-src http://http.us.debian.org/debian testing main contrib non-free
deb-src http://non-us.debian.org/debian-non-US testing/non-US main
contrib non-free
deb-src http://http.us.debian.org/debian unstable main contrib non-free
deb-src http://non-us.debian.org/debian-non-US unstable/non-US main
contrib non-free
```

Then get the source and make a local package:

```
$ apt-get source package/unstable
$ dpkg-source -x package.dsc
$ cd package-version
  ... inspect required packages (Build-depends in .dsc file) and
      install them too.  You need fakeroot too.

$ dpkg-buildpackage -rfakeroot

  ...or (no sig)
$ dpkg-buildpackage -rfakeroot -us -uc # use "debsign" later if needed

  ...or (no sig)
# ./debian/rules binary
# ./debian/rules clean
# cd ..
# dpkg-source -b package-version

  ...Then to install
# dpkg -i packagefile.deb
```

Usually, one needs to install a few packages with the “-dev” suffix to satisfy package dependencies. `debsign` is in the `devscripts` package. `auto-apt` may ease satisfying these dependencies.

In Woody, these dependency issues can be simplified. For example, to compile a source-only `pine` package:

```
# apt-get build-dep pine
# apt-get source -b pine
```

6.3.10. Convert or install an alien binary package

`alien` enables the conversion of binary packages provided in Redhat `rpm`, Stampede `slp`, Slackware `tgz`, and Solaris `pkg` file formats into a Debian `deb` package. If you want to use a package from another Linux distribution than the one you have installed on your system, you can use `alien` to convert it to your preferred package format and install it. It also supports LSB packages.

6.4. Other Debian peculiarities

6.4.1. dpkg-divert command

`dpkg-divert` forces `dpkg` not to install a file into its location, but to a *diverted* location.

System administrators or maintenance scripts can use this to override some package's configuration file, or whenever some files (which aren't marked as `conffiles`) need to be preserved by `dpkg`, when installing a newer version of a package which contains those files.

```
# dpkg-divert [--add]  filename # add "divert"
# dpkg-divert --remove filename # remove "divert"
```

6.4.2. `equivs` package

If you compile a program from source, it is best to make it into a real local debianized package (`*.deb`). Use `equivs` as a last resort.

```
Package: equivs
Priority: extra
Section: admin
Description: Circumventing Debian package dependencies
 This is a dummy package which can be used to create Debian
 packages, which only contain dependency information.
```

6.4.3. Alternative commands

To make the command `vi` run `vim`, use `update-alternatives`:

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Items in the Debian alternatives system are kept in `/etc/alternatives` as symlinks.

To set your favorite X window manager, use `x-window-manager` instead.

`/bin/sh` is a direct symlink to `/bin/bash` or `/bin/ash/.` It's safer to use `/bin/bash` to be compatible with old bashism-contaminated scripts but better discipline to use `/bin/ash` to enforce POSIX compliance.

6.4.4. System-V `init` and runlevels

The default runlevel to boot into can be set in `/etc/inittab`.

Unlike other distributions, Debian makes the management of runlevel completely the sysadmin's responsibility. Management of System-V style `init` on Debian is intended to be performed through `update-rc.d` scripts.

Starting `/etc/init.d/`*`name`* in runlevel 1,2,3 and stopping in 4,5 with sequencing priority number 20 (normal) can be done by:

```
# update-rc.d name start 20 1 2 3 . stop 20 4 5 .
```

Removing symbolic links while the script in `init.d` still exists can be done by:

```
# update-rc.d -f name remove
```

For editing runlevels, I cheat. I edit entries manually using the `mv` command at the shell prompt of `mc` while copying link entries using `Alt-Enter`. I even disable a daemon by inserting `exit 0` at the start of an `init.d` script as a quick hack. These are conffiles after all.

6.4.5. Disabled daemon services

Debian distribution cares system security seriously and expects the system administrator to be really a competent one. Thus, sometimes ease of use becomes secondary concern and many daemon services come with the highest security level with the least or no service available as their default install state.

Check `ps aux` or contents of `/etc/init.d/*` and `/etc/inetd.conf`, if you have doubt (exim, dhcp, …). Also check `/etc/hosts.deny` as 'Control de acceso through PAM and login' en la página 88.

X11 doesn't allow TCP/IP (remote) connections as default in the recent versions of Debian. See 'TCP/IP connection to X' en la página 98.

Capítulo 7

The Linux kernel under Debian

Debian has its own method of recompiling the kernel and related modules.

7.1. Kernel recompile

7.1.1. Debian standard method

Use the new `kernel-package` in unstable (7/2001). Also, some arguments to the `tar` command changed between Potato and Woody, so use `-j` instead of `-I` for `.bz2`. Also watch out for bug reports on `gcc`, `binutils`, and `modutils`.

Compiling a custom kernel from source under a Debian system requires special care. Use the new `--append_to_version` with `make-kpkg` to build multiple kernel-images.

```
# apt-get install debhelper modutils kernel-package libncurses5-dev
# apt-get install kernel-source-2.4.12   # use latest version
# vi /etc/kernel-pkg.conf                # input my name and email
$ cd /usr/src                            # build directory
$ tar -xvjf kernel-source-2.4.12.tar.bz2
$ cd kernel-source-2.4.12                # if this is your kernel source
$ rm -rf */pcmcia
    # [OPTIONAL] if one wants to use modules from pcmcia-cs or no pcmcia
$ cp /boot/config-2.4.12-386 .config     # get current config as default
$ make menuconfig                        # customize as one wishes
$ make-kpkg clean                        # must run (per: man make-kpkg)
$ fakeroot make-kpkg --append_to_version -486 --initrd \
        --revision=rev.01 kernel_image
```

```
$ cd ../modules/pcmcia
$ fakeroot ./debian/rules
$ cd ../..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # install
```

`make-kpkg kernel_image` actually does `make oldconfig` and `make dep`. Do not use `--initrd` if initrd is not used.

One can avoid `rm -fr */pcmcia` by selecting "General setup —>" to "PCMCIA/CardBus support —>" in `make menuconfig` and setting configuration as "< > PCMCIA/CardBus support".

7.1.2. Classic method

Get pristine sources from:
- Linux: http://www.kernel.org/
- pcmcia-cs: http://pcmcia-cs.sourceforge.net/

or use equivalent source in Debian and do the following:

```
# cd /usr/src
# tar xfvz linux-whatever.tar.gz
# rm -rf linux
# ln -s linux-whatever linux
# tar xfvz pcmcia-cs-whatever.tar.gz
# ln -s pcmcia-cs-whatever pcmcia
# cd linux
# rm -rf */pcmcia
    # [OPTIONAL] if one wants to use modules from pcmcia-cs or no pcmcia
# make menuconfig
... configure stuff ...
# make dep
# make bzImage
... edits for lilo / grub ...
... move /usr/src/linux/arch/i386/boot/bzImage to boot ...
... /sbin/lilo or whatever you do for grub
# make modules; make modules_install
# cd ../pcmcia
# make config
# make all
# make install
... add needed module names to /etc/modules
# shutdown -r now
... boot to new kernel ...
```

Use of `gcc`, `binutils`, and `modutils` from Debian unstable may help.

7.2. The modularized 2.4 kernel

The new Debian 2.4 kernels are very modularized. You have to make sure those modules are present to make the kernel function as you intend.

7.2.1. PCMCIA

`/etc/modules` needs to contain the following for PCMCIA to function:

```
# ISA PnP driver
isa-pnp
# Low level PCMCIA driver
# yenta_socket # does not seem to be needed im my case
```

The rest is taken care of by pcmcia scripts (from the `pcmcia-cs` package), `depmod` and `kmod`. I think I needed `isa-pnp` because my laptop is an old isa-pcmcia. Recent laptops with cardbus-pcmcia may not require this.

Voice of the generous Miquel van Smoorenburg `<miquels@cistron.nl>`

I simply removed the entire pcmcia stuff from the laptop here at work, including the cardmgr etc and just installed a 2.4 kernel with cardbus support, and the new “hotplug” package from Woody.

As long as you only have 32-bit cards you don’t need the pcmcia package; 2.4 has cardservices built in. And the standard tulip driver should work fine with your dlink card.

—Mike.

7.2.2. SCSI

[NOT TESTED] `/etc/modules` needs to contain the following for SCSI to function:

```
# SCSI core
scsi_mod
# SCSI generic driver
sg
# SCSI disk
sd_mod
# All other needed HW modules
...
```

`depmod` may take care of some of the above modules.

7.2.3. Network function

`/etc/modules` needs to contain the following for extra network function:

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (in order)
ip_tables
ip_conntrack
ip_conntrack_ftp
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
ipt_REDIRECT
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
#ipchains
```

```
#ipfwadm
```

The preceding may not be optimized. `depmod` may take care of some of the above modules.

7.2.4. EXT3 filesystem (> 2.4.17)

Enabling a journaling filesystem with the EXT3 FS involves the following steps using a Debian precompiled kernel-image (> 2.4.17):

```
# cd /etc; mv fstab fstab.old
# sed ’s/ext2/ext3,ext2/g’ <fstab.old >fstab
# vi /etc/fstab
... set root filesystem type to "auto" instead of "ext3,ext2"
# cd /etc/mkinitrd
# echo jbd >>modules
# echo ext3 >>modules
# echo ext2 >>modules
# cd /
# apt-get update; apt-get install kernel-image-2.4.17-686-smp
... install latest kernel and set up boot (lilo is run here)
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... For all EXT2 FS’s converted to EXT3
# shutdown -r now
```

Now EXT3 journaling is enabled. Using `ext3,ext2` as the `fstab` “type” entry ensures safe fallback to EXT2 if the kernel does not support EXT3 for non-root partitions.

If you have previously installed a 2.4 kernel and do not wish to reinstall, perform the above steps up to the `apt-get` commands, then:

```
# mkinitrd -o /boot/initrd.img-2.4.17-686-smp /lib/modules/2.4.17-686-smp
# lilo
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... for all EXT2 FS’s converted to EXT3
# shutdown -r now
```

Now EXT3 journaling is enabled.

If `/etc/mkinitrd/modules` was not set when `mkinitrd` was run and you would like to add some modules at boot time:

```
... at initrd prompt to gain shell (5 sec.), type RETURN
# insmod jbd
# insmod ext3 # modprobe ext3 may take care of everything
# insmod ext2
# ^D
... continue booting
```

At the system boot screen (`dmesg`), “cramfs: wrong magic” appears but this is known to be harmless. (This will be resolved in a later release.)

Some systems experience severe kernel lock-up if EXT3 is enabled (as of 2.4.17).

7.2.5. Realtek RTL-8139 support in 2.4

For whatever reason, the RTL-8139 support module is no longer called rtl8139, it’s now called 8139too. Just edit your `/etc/modules` to reflect this change when upgrading a 2.2 kernel to a 2.4 kernel.

Capítulo 8

Trucos para Debian

8.1. Arrancando el sistema

Véase el BootPrompt-HOWTO (LDP) para información detallada sobre el indicador de arranque.

8.1.1. "¡Olvidé la contraseña de superusuario!" (1)

Es posible arrancar el sistema y acceder a la cuenta de superusuario sin conocer la contraseña siempre y cuando se tenga acceso al teclado de la consola (esto supone que ni la BIOS ni un cargador de arranque como `lilo` solicitan una contraseña para evitar el arranque del sistema)

Este es un procedimiento que no requiere de discos de arranque externos ni cambios en los parámetros de arranque de la BIOS. Aquí, "Linux" hace referencia al kernel de Linux de la instalación predeterminada de Debian.

En el pantalla de arranque de `lilo`, cuando aparece `boot:` (en algunos sistemas debe presionar la tecla Mayús para evitar el arranque automático) escriba:

```
boot: Linux init=/bin/sh
```

Esto hará que el sistema arranque el kernel y ejecute el `/bin/sh` en vez del estándar `init`. Ahora tenemos privilegios de superusuario y acceso al intérprete de comandos. Puesto que `/` generalmente está montado en modo sólo lectura y aún no han sido montadas las distintas particiones, debe hacer lo siguiente para tener un sistema que funcione razonablemente.

```
init-2.03# mount -n -t remount,rw /
init-2.03# mount -avt nonfs,noproc,nosmbfs
```

```
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow
```

(si el segundo campo de datos en el `/etc/passwd` es una "x" para cada nombre de usuario, su sistema utiliza contraseñas ocultas y debe editar el archivo `/etc/shadow`) Para desactivar la contraseña de root, edite el segundo campo de datos en el archivo de contraseñas de modo que quede vacío. Ahora se puede reiniciar el sistema y entrar como root sin contraseña. A diferencia de algunas distribuciones Linux antiguas, la Debian actual (Potato) requiere de contraseña sólo cuando arranca en el nivel de ejecución 1.

8.1.2. "¡Olvidé la contraseña de superusuario!" (2)

Arranque desde un disco de emergencia. Si `/dev/hda3` es la partición raíz original, lo siguiente le permitirá editar el archivo de contraseñas tan fácilmente como antes.

```
# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd
```

La ventaja de este enfoque sobre el anterior es que no se necesita conocer la contraseña de `lilo` (si existe). Pero para usarlo uno debe ser capaz de poder acceder a la configuración de la BIOS para permitir arrancar el sistema desde un disquete o CD si es que ya no está configurada como tal.

8.1.3. No puedo arrancar el sistema

Si se tomó la molestia de crear un disco de arranque durante la instalación, no habrá ningún problema. Si `lilo` se encuentra dañado, grabe el disco de arranque desde el disco de instalación de debian y arranque el sistema con él. En el indicador de arranque, suponiendo que la partición raíz de su instalación Linux es `/dev/hda12` y que desea entrar al nivel de ejecución 3, escriba:

```
boot: rescue root=/dev/hda12 3
```

De esta manera arrancará con un sistema prácticamente funcional usando el kernel del disquete (pueden existir pequeños inconvenientes debido a la falta de algunas características o módulos del kernel).

Si necesita un disquete de arranque personalizado, consulte el `readme.txt` sobre el disco de rescate.

8.1.4. Otros trucos con el indicador de arranque

El sistema se puede arrancar en un nivel de ejecución particular y con una configuración determinada usando el indicador de arranque de `lilo`. Para más detalles consulte el BootPrompt-HOWTO (LDP).

Si desea arrancar el sistema en el nivel de ejecución cuatro, use la siguiente entrada para el indicador de arranque de `lilo`.

```
boot: Linux 4
```

Si desea arrancar el sistema en modo monousuario y conoce la contraseña de superusuario, algunos de los siguientes ejemplos funcionará usando el indicador de arranque de `lilo`.

```
boot: Linux S
boot: Linux 1
boot: Linux -s
```

Si desea arrancar el sistema con menos memoria de la que realmente posee el sistema (por ejemplo, 48MB en un sistema con 64MB), use la siguiente entrada para el indicador de arranque de `lilo`:

```
boot: Linux mem=48M
```

Asegúrese de no especificar un valor mayor al tamaño de memoria real ya que si es así el kernel se colgará. Los núcleos antiguos `y/o` las placas madres con una BIOS antigua no usan la memoria por encima de los 64 MB, salvo que uno escriba `mem=128M` en el indicador de arranque o incluya una línea similar en el `/etc/lilo.conf`.

8.2. Registro de actividades

8.2.1. Registrando las actividades del intérprete de comandos

La administración del sistema implica tareas mucho más elaboradas en un entorno Unix que un entorno común de una computadora personal. Asegúrese de saber lo básico sobre configuración por si acaso necesita recuperar el sistema de algún problema. Las herramientas de configuración con GUI basadas en X-Window son agradables y convenientes pero a menudo inapropiadas en situaciones de emergencia.

En este contexto, el registro de las actividades del intérprete de comandos resulta ser una buena práctica especialmente como superusuario.

Emacs: utilice `M-x shell` para empezar a grabar en el búfer y `C-x C-w` para escribir el contenido del mismo en un archivo.

Intérprete de comandos: utilice el comando `script`.

```
$ script
Script started, file is typescript
 ... hacer cualquier cosa ...
 Control-D
$ col -bx <typescript >savefile
$ vi savefile
```

Se puede utilizar lo siguiente en vez del comando `script`:

```
$ bash -i 2>&1 | tee typescript
```

8.2.2. Registrando las actividades en X

Si necesita registrar la imagen de una aplicación X, incluyendo una terminal xterm, utilice `gimp` (GUI). Puede capturar cada ventana o la totalidad de la pantalla. Otras alternativas son `xwd` (`xbase-clients`), `import` (`imagemagick`) o `scrot` (`scrot`).

8.3. Copiar y archivar un subdirectorio entero

8.3.1. Comandos básicos para copiar un subdirectorio entero

Si necesita reordenar la estructura de archivos, mueva el contenido incluyendo los enlaces a archivos mediante:

```
Método estándar:
# cp -a /directorio/fuente /directorio/destino # debe ser GNU
# (cd /directorio/fuente && tar cf - . ) | \
        (cd /directorio/destino && tar xvfp - )
Si existen enlaces duros, se necesita un método más elaborado:
# cd /ruta/al/directorio/original
# find . -depth -print0 | afio -p -xv -0a
# /punto/de/montaje/del/nuevo/directorio
En el caso de una conexión remota:
# (cd /directorio/fuente && tar cf - . ) | \
```

```
     ssh usuario@host.dom (cd /directorio/destino && tar xvfp - )
Si no hay archivos enlazados:
# scp -pr usuario1@host1.dom:/directorio fuente \
          usuario2@host2.dom:/directorio/destino
```

En este caso, `scp` <==> `rcp` y `ssh` <==> `rsh`.

El método para copiar un subdirectorio entero está basado en la información proporcionada por Manoj Srivastava <srivasta@debian.org> de la lista debian-user@lists.debian.org.

8.3.2. `cp`

Inicialmente, cp no resultaba ser el candidato correcto ya que no desreferenciaba enlaces simbólicos ni tampoco preservaba enlaces duros. Otra cosa a considerar eran los archivos de tamaño muy pequeño.

GNU cp ha superado estas limitaciones. No obstante en sistemas que no son GNU cp puede aún tener problemas. Asimismo, no se pueden generar archivos portables pequeños usando cp.

```
% cp -a . directorio_nuevo
```

8.3.3. `tar`

Tar soluciona algunos de los problemas que tenía cp con los enlaces simbólicos. Sin embargo, 'cpio' maneja archivos especiales que el 'tar' tradicional no puede manejar.

'tar' maneja múltiples enlaces duros colocando una única copia de un enlace en la cinta. El nombre asignado a dicha copia es el *único* que se puede usar para recuperar el archivo. En cambio, 'cpio' coloca una copia de cada enlace de manera que para recuperar el archivo se puede utilizar el nombre de cualquiera de ellas.

8.3.4. `pax`

Es la nueva utilidad para archivar, portable y compatible POSIX (IEEE Std 1003.2-1992, páginas 380-388 (sección 4.48) y páginas 936-940 (sección E.4.48)). Pax lee, escribe y lista los componentes de un directorio y copia la jerarquía de los mismos. La operación pax es independiente del formato específico del archivo y admite una amplia variedad de formatos.

Las implementaciones de pax son aún nuevas y están en pleno desarrollo.

```
$ pax -rw -p e . directorio_nuevo
 o
$ find . -depth  | pax -rw -p e  directorio_nuevo
```

8.3.5. `cpio`

Cpio almacena o extrae archivos en o de un archivo tar o cpio. El archivo puede ser otro archivo del disco, una cinta magnética o una tubería.

```
$ find . -depth -print0 | cpio --null --sparse -pvd new-dir
```

8.3.6. `afio`

Afio es una mejor forma de tratar con archivos con formato cpio. Generalmente es más rápido que cpio, proporciona más opciones para utilizar con cintas magnéticas y maneja mejor la corrupción de los datos de entrada. Admite archivos multi-volumen durante su operación interactiva. Afio puede crear archivos empaquetados que son mucho más seguros que los empaquetados con tar o cpio. En un script, para realizar una copia de seguridad, afio tiene un comportamiento óptimo como 'motor para archivar'.

```
$ find . -depth -print0 | afio -px -0a directorio_nuevo
```

Para crear todas mis copias de respaldo en una cinta uso afio.

8.4. Modificar archivos con la sustitución de expresiones regulares

Para reemplazar todas las instancias FROM_REGEX por TO_REGEX en todos los archivos FILES ...:

```
# perl -i -p -e 's/FROM_REGEX/TO_REGEX/g;' FILES ...
```

`-i` es para "editar en el lugar", `-p` significa "bucle implícito sobre los archivos FILES ...". Si la sustitución es compleja, la recuperación ante posibles errores resulta más fácil usando el parámetro `-i.bak` en vez de `-i`. De esta manera se conservará una copia de los archivos originales a los que se les agregará la extensión `.bak`.

8.5. Recuperar al sistema de un cuelgue

8.5.1. Mate el proceso

Ejecute `top` para ver cuál es el proceso que está actuando de manera extraña. Pulse "P" para ordenar por tiempo de cpu, "M" para ordenar por uso de memoria y "k" para matar un proceso.

Utilice `kill` junto el ID del proceso para matar (o enviar una señal a) un proceso, `killall` para hacer lo mismo pero usando el nombre del comando. Señales de uso habitual:

```
 1: HUP,  reiniciar demonio
15: TERM, terminar un proceso en forma normal
 9: KILL, matar un proceso sin contemplaciones
```

8.5.2. ALT-SysRq

La opción de compilación del kernel "Magic SysRq key" proporciona una protección contra el mal funcionamiento del sistema. Pulsando `ALT-SysRq` en una i386 y a continuación una de las teclas `r 0 k e i s u b` se obtiene el pase mágico.

Un`R`aw restaura el teclado tras el cuelgue de X. Modifique el nivel de registro de la consola con `0` para reducir los mensajes de error. sa`K` (tecla de atención del sistema) mata a todos los procesos en la consola virtual actual. t`E`rminate mata a todos los procesos de la terminal actual salvo init. k`I`ll mata a todos los procesos incluyendo a init.

`S`ync (sincronizar), `U`mount (desmontar), y re`B`oot (reiniciar) a menudo se utilizan en el caso de situaciones realmente complicadas.

En el momento de escribir este documento, los núcleos que vienen con la instalación predeterminada de Debian no han sido compilados con esta opción. Recompile el kernel para activarla. Se puede encontrar información detallada en: `/usr/share/doc/kernel-doc-version/Documentation/sysrq.txt.gz`.

8.6. Archivos de configuración

8.7. Algunos pequeños comandos útiles para tener en cuenta

8.7.1. Memoria disponible

`free` y `top` brindan una buena información sobre los recursos de memoria disponibles. No se preocupe por el tamaño que figura bajo "used" de la línea "Mem:" sino por el valor que se encuentra justo debajo de él (38792 en el siguiente ejemplo).

```
$ free -k # para una máquina con 256MB
             total       used       free     shared    buffers cached
Mem:        257136     230456      26680      45736     116136 75528
-/+ buffers/cache:      38792     218344
Swap:       264996          0     264996
```

La cantidad de memoria física exacta se puede confirmar haciendo `grep '^Memory' /var/log/dmesg`. En este caso arroja el siguiente resultado: "Memory: 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init)".

```
Total         = 262144k = 256M (1k=1024, 1M=1024k)
Free to dmesg = 256984k = Total - kernel - reserved - data - init
Free to shell = 257136k = Total - kernel - reserved - data
```

El sistema no puede usar cerca de 5MB ya que lo utiliza el propio kernel.

8.7.2. Configurar fecha y hora (BIOS)

```
# date MMDDhhmmCCYY
# hwclock --utc
# hwclock --systohc
# hwclock --show
```

Esto fijará la fecha y hora del sistema y del hardware en MM/DD hh:mm, CCYY. La hora se muestra según el huso horario local pero el hardware utiliza el UTC.

Configure el reloj del sistema para corregir la fecha y hora en forma automática mediante un servidor remoto:

```
# ntpdate <server>
```

Si su sistema posee una conexión a Internet permanente, resulta interesante incluirlo en `/etc/cron.daily`.

8.7.3. Como desactivar el protector de pantalla

En la consola de Linux:

```
# setterm -powersave off
```

Ejecute la consola kon2 (kanji) con:

```
# kon -SaveTime 0
```

Cuando esté ejecutando X:

```
# xset s off
 o
# xset -dpms
 o
# xscreensaver-command -prefs
```

Consulte las correspondientes páginas del manual.

8.7.4. Desactivar el sonido (bip)

Uno siempre puede desenchufar el parlante del PC ;-) Para el intérprete de comandos:

```
echo "set bell-style none">> ~/.inputrc:
```

8.7.5. Mensajes de error en la pantalla

El primer lugar que hay que consultar para leer los mensajes de error que aparecen por pantalla es `/etc/init.d/klogd`. En este script haga la siguiente asignación `KLOGD="-c 4"` y ejecute `/etc/init.d/klogd restart`.

Un método alternativo consiste en ejecutar `dmesg -n1`.

Otro lugar que hay que ver es el `/etc/syslog.conf`.

8.7.6. Volver la consola a su estado normal

Si la pantalla se enloquece después de hacer `$ cat archivo_binario` (quizás no pueda ver el comando que ingresa mientras escribe) haga:

```
$ reset
```

8.7.7. Convertir un archivo de texto en formato DOS a formato Unix

Convertir un archivo de texto en formato DOS (fin-de-línea=^M^J) en un archivo Unix (^J).

```
# apt-get install sysutils
$ dos2unix dosfile
```

8.7.8. Convertir un archivo grande en archivos más pequeños

```
$ split -b 650m archivo   # dividir el archivo en partes de 650 MB
$ cat x* >archivo_grande  # unir los archivos en un archivo grande
```

8.7.9. Pequeños scripts que incluyen redireccionamientos

Los siguientes scripts realizan tareas útiles utilizando tuberías.

```
find /usr | egrep -v "/usr/var|/usr/tmp|/usr/local"
                      # encontrar todos los archivos en /usr
      # excluyendo algunos archivos
xargs -n 1 command    # ejecutar comando para tods los ítems de la stdin
xargs -n 1 echo|      # escribir ítems separados por espacios en blanco
                      # en renglones
grep -e patrón|       # extraer líneas con un determinado patrón
cut -d: -f3 -|        # extraer el tercer campo separado por :
                      # (archivo passwd, etc.)
col -bx |             # eliminar retrocesos y convertir tabulaciones en
                      # espacios
expand -|             # transformar las tabulaciones
sort -u|              # ordenar y eliminar duplicados
tr '\n' ' '|          # concatenar líneas en una sola línea
tr '\r' ''|           # eliminar CR (retornos de carro)
tr 'A-Z' 'a-z'|       # convertir mayúsculas en minúsculas
sed 's/^/# /'|        # transformar cada línea en un comentario
sed 's/\.ext//g'|     # eliminar .ext
sed  -n -e 2p|        # mostrar la segunda línea
head -n 2 -|          # mostrar las primeras 2 líneas
tail -n 2 -|          # mostrar las últimas 2 líneas
```

8.7.10. Obtener el texto de una página web o del archivo de una lista de correos

Lo siguiente leerá una página web como un archivo de texto. Resulta muy útil cuando se desea copiar configuraciones que se obtienen de la red.

```
$ lynx -dump http://www.remote-site.com/help-info.html >archivo_de_texto
```

Si se trata de un archivo de una lista de correo, use `munpack` para obtener los contenidos mime del texto.

8.7.11. El tiempo de un comando

Mostrar el tiempo empleado por un proceso.

```
# time df >/dev/null
real    0m0.035s       # tiempo de reloj (tiempo real transcurrido)
user    0m0.000s       # tiempo en modo usuario
sys     0m0.020s       # tiempo en modo kernel
```

8.7.12. El comando `nice`

Use `nice` (del paquete GNU shellutils) para fijar el valor "nice" de un comando al ejecutarlo. `renice` (bsdutild) o `top` puede modificar el valor "nice" de un proceso. El proceso más lento tiene el valor 19 (prioridad más baja); los valores negativos son "not-nice". El valor -20 lo tiene el proceso más veloz (prioridad alta). Sólo el superusuario puede fijar valores "nice" negativos.

```
# nice  -19 top                                         # muy lento
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img # muy rápido
```

8.7.13. Planificar una actividad (`cron`, `at`)

Use `cron` y `at` para planificar tareas en Linux. Véase `at(1),crontab(5),crontab(8).`

Ejecute el comando `crontab -e` para crear o editar el archivo crontab para configurar eventos planificados. Ejemplo de un archivo crontab:

```
# utilice /bin/sh para ejecutar los comandos sin importar lo que dice el
# /etc/passwd
SHELL=/bin/sh
# envíe un mensaje a 'pablo' sin importar a quien pertenece el crontab
MAILTO=pablo
# Minuto Hora Día_del_mes Mes Día_de_la_semana comando
# ejecutar todos los días a las 00:05
5  0  *  * *   $HOME/bin/tarea.diaria >> $HOME/tmp/salida 2>&1
# ejecutar a las 14:15 el primer día de cada mes -- enviar salida a Pablo
15 14 1  * *   $HOME/bin/mensual
# ejecutar a las 22:00 todos los días hábiles (1-5), molestar a José.
# % para nueva línea, el último % para cc:
0 22 *   * 1-5 mail -s "Son las 10 de la noche" josé%José:%%¿Dónde están
                        los chicos?%.%%
```

```
23 */2 1 2 *   echo "ejecutar el 1 de febrero a los 23 minutos después
                     de 0am, 2am, 4am ..."
5  4 *   * sun echo "ejecutar todos los domingos a las 04:05"
# ejecutar a las 03:40 el primer lunes de cada mes
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && comando -args
```

Ejecutar el comando `at` para planificar una tarea una sola vez:

```
$ echo 'command -args'| at 3:40 monday
```

8.7.14. Intercambiando consolas con **`screen`**

El programa `screen` permite ejecutar múltiples terminales virtuales, cada una con su intérprete de comandos interactivo, en una única terminal física o ventana que emule un terminal. Incluso si utiliza consolas virtuales Linux o múltiples ventanas xterm, merece la pena experimentar con `screen` por sus amplias funcionalidades que incluye navegación por el histórico de comandos, copiar y pegar, registro de accesos al sistema, entrada de caracteres especiales y la capacidad de separar una sesión entera de `screen` del terminal para recuperarla posteriormente. Si frecuentemente se conecta a una máquina Linux desde un terminal remoto o usando el terminal VT100, `screen` le facilitará muchísimo las cosas.

Una vez que arranca `screen` toda la entrada de datos que se hace a través del teclado se envía a la ventana actual excepto la combinación de teclas de comando, por defecto `^A`. Todos los comandos de `screen` se ingresan escribiendo `^A` seguido de una sola tecla [y eventualmente algunos parámetros]. Algunos comandos útiles:

```
^A ?     mostrar ventana de ayuda (muestra las asociaciones de teclas)
^A c     crear una nueva ventana y cambiar a ella
^A n     ir a la siguiente ventana
^A p     ir a la ventana anterior
^A 0     ir a la ventana número 0
^A w     mostrar la lista de ventanas
^A a     enviar un Ctrl-A a la ventana actual como entrada del teclado
^A h     grabar una copia de la ventana actual a un archivo
^A H     comenzar/finalizar la grabación de ventana actual en un archivo
^A ^X    bloquear la terminal (protegida con contraseña)
^A d     separar una sesión del terminal
^A DD    separar una sesión y salir
```

La posibilidad de `screen` de "separar" una sesión es muy poderosa. Supongamos que estamos conectados a través de una conexión telefónica y estamos ejecutando una sesión compleja con `screen` que incluye editores y otros programas abiertos en diversas ventanas. De repente, necesitamos abandonar el terminal pero no queremos perder nuestro trabajo al colgar la línea. Simplemente, escriba `^A d` para separar la sesión y luego desconéctese (o, más rápido aún, escriba `^A DD` para desprender la sesión de `screen` y simultáneamente desconectarse). Cuando se conecte nuevamente, escriba el comando `screen -r` y `screen` automáticamente recuperará todas las ventanas que había abierto.

Éste es sólo una pequeña muestra de los comandos y características de `screen`. Si hay algo que desea que `screen` haga, ¡existe una gran posibilidad que pueda hacerlo!. "man screen" para más detalles.

Observación: si ve que la tecla de retroceso y/o Ctrl-H no funciona correctamente cuando ejecuta `screen`, edite el archivo `/etc/screenrc`, localice la línea

```
bindkey -k kb stuff "\177"
```

y coméntela (es decir, agréguele el carácter "#" al principio).

8.7.15. Probando la red

Instale `netkit-ping`, `traceroute`, `dnsutils`, `ipchains` (para el kernel 2.2), `iptables` (para el kernel 2.4) y el paquete `net-tools`:

```
$ ping yahoo.com            # verificar la conexión a Internet
$ traceroute yahoo.com      # rastrear paquetes IP
$ ifconfig                  # verificar la configuración del
                            # anfitrión (host)
$ route -n                  # verificar la configuración de la ruta
$ dig [@dns-server.com] host.dom [{a|mx|any}] |less
      # verificar registros host.dom DNS [@ dns-server.com] para
      # un registro [{mx|any}]
$ ichains  -L -n |less      # verificar filtrado de paquetes (kernel 2.2)
$ iptables -L -n |less      # verificar filtrado de paquetes (kernel 2.4)
$ netstat -a                # mostrar todos los puertos abiertos
$ netstat -l --inet         # mostrar los puertos en escucha
$ netstat -ln --tcp         # mostrar puertos tcp en escucha (numérico)
```

8.7.16. Eliminar mensajes de la cola local

Para eliminar los mensajes de la cola local:

```
# exim -q    # eliminar mensajes en espera
# exim -qf   # eliminar todos los mensajes
# exim -qff  # eliminar incluso mensajes bloqueados
```

“-qff” puede resultar mejor para el script /etc/ppp/ip-up.d/exim.

8.7.17. Eliminar mensajes bloqueados de la cola local

Para eliminar mensajes de correo bloqueados de la cola local reenviando el mensaje de error:

```
# exim -Mg `mailq | grep frozen | awk '{ print $3 }'`
```

8.7.18. Borrar el contenido de un archivo

No utilice `rm` si el archivo es usado por otros usuarios.

```
$ :>archivo-a-eliminar
```

8.7.19. Archivos fantasma

```
$ dd if=/dev/zero  of=nombre_archivo bs=1k count=5 # 5KB de contenido
                                                   # cero
$ dd if=/dev/urandom of=nombre_archivo bs=1m count=7 # 7MB de contenido
                                                     # al azar
$ touch nombre_archivo      #  crear un archivo de 0B o actualizar mtime
                            #  (fecha de la última modificación)
```

8.7.20. chroot

Puede crear un sistema independiente que comparta el mismo kernel.

```
# mount /dev/hda1 /mnt/target
 ... suponiendo que /dev/hda1 contiene un sistema entero
# chroot /mnt/target
 ... Ahora los contenidos de /dev/hda1 se ven como el directorio raíz.
# mount proc /proc             # por si acaso
 ... ejecute el comando dentro de /dev/hda1
```

Esto permite cargar las ramas estable/de prueba/inestable en una misma máquina. Asimismo, uno puede ejecutar un programa que requiera gran cantidad de memoria (por ejemplo, `dselect`) en una máquina anfitrión, montando un máquina auxiliar (máquina satélite) mediante NFS en modo lectura/escritura y accediendo a ella mediante `chroot`.

En woody, el sistema chroot puede crearse fácilmente con el comando `debootstrap`.

```
# mkdir potatochroot
# debootstrap potato potatochroot
# chroot potatochroot
# apt-setup # configura el /etc/apt/sources.list
```

8.7.21. Samba

Lo básico para obtener archivos desde Windows:

```
# mount -t smbfs -o username=myname,uid=my_uid,gid=my_gid \
        //server/share /mnt/smb  # monta un directorio Windows en Linux
# smbmount //server/share /mnt/smb \
        -o "username=mi_nombre,uid=mi_uid,gid=mi_gid"
# smbclient -L 192.168.1.2 # lista los archivos compartidos de una
                           # máquina
```

Capítulo 9

Tuning a Debian system

This chapter only describe basics of system configuration. Prerequisite of this chapter is reading of 'Debian System installation hints' en la página 27.

Also for the security concious, it is highly recommended to read Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`) which can also be found as `harden-doc` package.

9.1. System initialization hints

See 'The `init` program' en la página 22 for the basics of Debian init script.

9.1.1. Customizing init script

Debian uses sys-V init sctript system. Although all these init scripts in `/etc/init.d/*` are marked as conffiles and sysadmins are free to modify them, customizing these init scripts through editting files in `/etc/defaults/*` are preffered approach.

For example `/etc/init.d/rcS` can customize boot time defaults for `motd`, `sulogin`, ...

9.1.2. Customizing system logging

System log mode can be configured `/etc/syslog.conf`. Check the `colorize` package for a program to colorize system log files. See `syslogd(8)` and `syslog.conf(5)`.

9.1.3. Hardware access optimization

There are few hardware optimization configurations which Debian leaves them to the sysadmin to take care.

- `hdparm`
  - Hard disk access optimization. Very effective.
  - Dangerous. You must read `hdparm(8)` first.
  - `hdparm -tT` *`/dev/hda`* to test disk access spead.
  - `hdparm -c1 -d1 -u1 -m16 -A` *`/dev/hda`* to spead up modern IDE system. (It may be dangerous.)

- `setserial`
  - Collection of tools for serial port management.

- `scsitools`
  - Collection of tools for scsi hardware management.

- `memtest86`
  - Collection of tools for memory hardware management.

- `hwtools`
  - Collection of tools for low-level hardware management.
    - irqtune: changes the IRQ priority of devices to allow devices that require high priority and fast service (e.g. serial ports, modems) to have it.
    - scanport: scans `i/o` space from 0x100 to 0x3ff looking for installed ISA devices.
    - inb: a quick little hack that reads an `i/o` port and dumps the value in hex and binary.

- `schedutils`
  - Linux scheduler utilities.
  - `taskset`, `irqset`, `lsrt`, and `rt` are included.
  - Together with `nice` and `renice` (not included), they allow full control of process scheduling parameters.

9.2. Access control

9.2.1. Control de acceso through PAM and login

PAM provide login control.

```
/etc/pam.d/*              # PAM control files
/etc/pam.d/login          # PAM controla el acceso
/etc/security/*           # PAM module parameters
/etc/securetty            # controla el acceso del root por consola (login)
/etc/login.defs           # this controls login behaviors (login)
```

Bajo su propia responsabilidad, modifique el contenido del archivo `/etc/pam.d/login` de la siguiente manera, si desea acceder a la consola en forma insegura sin contraseña alguna.

```
#auth       required   pam_unix.so nullok
auth       required   pam_permit.so
```

Similar tricks can be applied for `xdm`, `gdm`, . . . for passwordless console X.

The maximum number of processes can be set with `ulimit -u 1000` in a Bash shell or with settings in `/etc/security/limits.conf` from PAM. Other parameters such as `core` can be set similarly. Initial value of `PATH` can be set by `/etc/login.defs` before the shell start up script.

The documentation for PAM is packaged in the `libpam-doc` package. The *Linux-PAM System Administrator's Guide* covers configuring PAM, what modules are available etc. The documentation also includes *The Linux-PAM Application Developers' Guide* and *The Linux-PAM Module Writers' Guide*.

9.2.2. Why GNU `su` does not support the `wheel` group

This is the famous phrase at the bottom of `info su` page by Richard M. Stallman. This section is there for historical reason only and we should not be fooled. Current `su` in Debian uses PAM thus can restrict ability to use `su` to any group using `pam_wheel.so` in `/etc/pam.d/su`. Following will set `adm` group in Debian system as an equivalent of BSD `wheel` group and allow `su` without password for them.

```
# anti-RMS configuration in /etc/pam.d/su
auth       required   pam_wheel.so group=adm

# Wheel members to be able to su without a password
auth       sufficient pam_wheel.so trust group=adm
```

9.2.3. Meaning of group

Few interesting groups:

- `root` group is default wheel group for `su` if `pam_wheel.so` is used without the `group=` argument.
- `adm` group can read logfiles.
- `cdrom` group can be used locally to give a set of users access to a cdrom drive.
- `floppy` group can be used locally to give a set of users access to a floppy drive.
- `audio` group can be used locally to give a set of users access to an audio device.
- `src` group owns source code, including files in `/usr/src`. It can be used locally to give a user the ability to manage system source code.
- `staff` is useful for more helpdesk/junior sysadmins type of people and gives them the ability to do things in `/usr/local` and create directories in `/home`.

For complete list, see "FAQ" section in Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`) which can also be found as `harden-doc` package.

9.2.4. `sudo` - safer work environment

My usage of `sudo` is mostly a protection from my own stupidity. I consider using `sudo` a better alternative to always using the system as root. YMMV.

Install `sudo` and activate it by setting options in /etc/sudoers (`examples/`). Also check out `sudo` group feature in `/usr/share/doc/sudo/OPTIONS`.

The sample configuration provides staff group members access to any commands run as root under `sudo` and also gives src members access to selected commands run as root under `sudo`.

The advantage of `sudo` is that it only requires an ordinary user's password to log in, and activity is monitored. This is nice way to give some authority to a junior administrator. For example:

```
$ sudo chown -R myself:mygrp .
```

Of course if you know the root password (as most home users do), any command can be run under root from a user account:

```
$ su -c "shutdown -h now"
Password:
```

(I know I should tighten the admin account's `sudo` privileges. But since this is my home server, I have not bothered yet.)

For a different program that allows ordinary users to run commands with root privileges, see the `super` package.

9.2.5. Access control to daemon programs

Internet *super-server*, `inetd` is started at boot time by `/etc/rc2.d/S20inetd` (for RUNLEVEL=2) which is symlink to `/etc/init.d/inetd`. Essentially, `inetd` allows running one daemon to invoke several others, reducing load on the system.

Whenever a request for service arrives, its protocol and service is identified by looking up database in `/etc/protocols` and `/etc/services`. For the normal internet service, `inetd` looks up database in `/etc/inetd.conf`. For Sun-RPC based service, `inetd` looks up database in `/etc/rpc.conf`.

For the system security, make sure to disable unused services in `/etc/inetd.conf`. Sun-RPC services need to be active for NFS and other RPC-based programs.

Sometimes, `inetd` does not start the intended server directry but starts `tcpd` tcp/ip daemon wrapper program with the intended server name as its argument in `/etc/inetd.conf`. In this case, `tcpd` runs the appropriate server program after `tcpd` logs the request and does some additional checks using `/etc/hosts.deny` and `/etc/hosts.allow`.

Comente “ALL: PARANOID” en el archivo `/etc/hosts.deny` si es que existe.

For details, see `inetd(8)`, `inetd.conf(5)`, `protocols(5)`, `services(5)`, `tcpd(8)`, `hosts_access(5)`, and `hosts_options(5)`.

For more information on Sun-RPC, see `rpcinfo(8)`, `portmap(8)`, and `/usr/share/doc/portmap/portmapper.txt.gz`.

9.2.6. Lightweight Directory Access Protocol

References:

- OpenLDAP (`http://www.openldap.org/`)
- OpenLDAP Admin Guide in `openldap-guide` package
- LDP: LDAP Linux HOWTO (`http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html`)
- LDP: LDAP Implementation HOWTO (`http://www.tldp.org/HOWTO/LDAP-Implementation-HOWTO` `index.html`)
- OpenLDAP, extensive use reports (`http://portal.aphroland.org/~aphro/ldap-docs/ldap.html`)
- Open LDAP with Courier IMAP and Postfix (`http://annapolislinux.org/docs/plc/postfix-courier-howto.txt`)

9.3. CD-writer

CD-writers with ATAPI/IDE interfaces have recently become a very popular option. It is a nice media for system backup and archiving for the home user needing < 640 MB capacity. For the most authoritative information, see the LDP CD-Writing-HOWTO (`http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html`).

9.3.1. Introduction

First, any disruption of data sent to the CD-writer will cause irrecoverable damage to the CD. Get a CD-writer with as large a buffer as possible. If money is no object, do not bother with ATAPI/IDE, just get a SCSI version. If you have a choice of IDE interface to be connected, use the one on the PCI-bus (i.e., on the motherboard) rather than one on the ISA-bus (an SB16 card, etc.).

When a CD-writer is connected to IDE, it has to be driven by the IDE-SCSI driver instead of an ordinary IDE CD driver. Also, the SCSI generic driver needs to be activated. There are two possible approaches for of doing this, assuming a kernel distributed with modern distributions (as of March 2001).

9.3.2. Approach 1: modules + `lilo`

Add the following line to `/etc/lilo.conf` if you are using a stock Debian kernel. If multiple options are used, list them separated by spaces:

```
append="hdx=ide-scsi ignore=hdx"
```

Here the location of the CD-writer, which is accessed through the ide-scsi driver, is indicated by `hdx`, where *x* represents one of the following:

```
hdb      for a slave on the first IDE port
hdc      for a master on the second IDE port
hdd      for a slave on the second IDE port
hde-hdh  for a drive on an external IDE port
```

Type the following commands as root to activate after finishing all the configuration:

```
# lilo
# shutdown -h now
```

9.3.3. Approach 2: recompile the kernel

Debian uses `make-kpkg` to create a kernel. Use the new `--append_to_version` with make-kpkg to build multiple kernel images. See 'The Linux kernel under Debian' en la página 65.

Use the following setup through `make menuconfig`:

- bzImage
- Exclude the IDE CD driver (not a must, but simpler to do this)
- Compile in ide-scsi and sg, or make them modules

9.3.4. Post-configuration steps

Kernel support for the CD-writer can be activated during booting by the following:

```
# echo ide-scsi >>/etc/modules
# echo sg       >>/etc/modules
# cd /dev; ln -sf scd0 cdrom
```

Manual activation can be done by:

```
# modprobe ide-scsi
# modprobe sg
```

After reboot, you can check installation by:

```
$ dmseg|less
# apt-get install cdrecord
# cdrecord -scanbus
```

[Per Warren Dodge] Sometines, there may be some conflicts between `ide-scsi` and `ide-cd` if there are both CD-ROM and CD-R/RW on the system. Try adding following line to your `/etc/modutils/aliases`, run `update-modules` and reboot may help.

```
pre-install      ide-scsi      modprobe ide-cd
```

This causes the IDE driver to load before `ide-scsi`. The IDE driver `ide-cd` takes control of the ATAPI CD-ROM — anything that it hasn't been told to **ignore**. That leaves just the ignored devices for `ide-scsi` to control.

9.3.5. CD-image file (bootable)

To create a CD-ROM of files under `target-directory/` as `cd-image.raw` (bootable, Joliet TRANS.TBL-enabled format; if not bootable, take out `-b` and `-c` options), insert a boot floppy in the first floppy drive and

```
# dd if=/dev/fd0 target-directory/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
        -o cd-image.raw target_directory/
```

One funny hack is to make a bootable DOS CD-ROM. If an ordinary DOS boot floppy disk image is in the above *boot.img*, the CD-ROM will boot as if a DOS floppy were in the first floppy drive (A:). Doing this with freeDOS may be more interesting.

This CD-image file can be inspected by mounting it on the loop device.

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

9.3.6. Write to the CD-writer (R, R/W):

First test with (assuming double speed)

```
# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Then if OK, write to CD-R with

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Or write to a CD-RW disk with

```
# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

Some CD-RW drives work better with

```
# nice --10 cdrecord -v blank=all speed=2 dev=0,0
```

followed by

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Two steps are needed to prevent SCSI timeouts during blanking from interfering with the burning step. The argument value to the `nice` may require some adjustments.

9.3.7. Make an image file of a CD

Some CD-R's and commercial CD's have junk sectors at the end that make copying by `dd` impossible (the Windows 98 CD is one of them). The `cdrecord` package comes with the `readcd` command. Use this to copy any CD contents to an image file. If it is a data disk, mount it and run `mount` to see its actual size. Divide the number shown (in blocks, = 1024 bytes) by 2 to get the number of actual CD sectors (2048 bytes). Run `readcd` with options and use this disk image to burn the CD-R/RW.

```
# readcd target lun scsibusno # select function 11
```

Here, set all 3 command line parameters to 0 for most cases. Sometimes the number of sectors given by `readcd` is excessive! Use the above number from an actual mount for better results.

```
My CD-R        = +2 sectors
MS Windows CD = +1 sector, i.e., +2048 bytes
```

9.3.8. Debian CD image

To obtain the latest information on Debian CD, visit the Debian CD site. (`http://www.debian.org/CD/`)

If you have fast Internet connection, think about installing over network using:

- a few floppy images (`http://www.debian.org/distrib/floppyinst`).
- a minimal bootable CD image (`http://www.debian.org/CD/netinst/`).

If you do not have fast Internet connection, think about purchasing CD's from a CD venders (`http://www.debian.org/CD/vendors/`).

Please do not waste bandwidth by downloading standard CD images unless you are a CD image tester (even with new jigdo-method).

One note worthy CD image is KNOPPIX - Live Linux Filesystem On CD (`http://www.knopper.net/knoppix/index-en.html`). This CD will boot functioning Debian system without installing itself to the hard disk.

9.3.9. Backup the system to CD-R

Copy key configuration files and data files to CD-R. Use script here: backup (`examples/`)

9.3.10. Copy a music CD to CD-R

Not tested by me:

```
# apt-get get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

or,

```
# apt-get install cdrdao #disk at once
# cdrdao read-cd --device /dev/cdrom --driver generic-mmc \
        --paranoia-mode 3 my_cd # read cd
# cdrdao write --device /dev/cdrom --driver generic-mmc --speed 8 \
        my_cd # write a new CD
```

`cdrdao` does a real copy (no gaps, etc. . .)

9.4. The X program

'X server' en la página siguiente a program on a local host that displays an X window and/or desktop on a user's monitor (CRT, LCD) and accepts keyboard and mouse input.

'X client' en la página 98 a program on a (local or remote) host that runs X-window-compatible application software.

This reverses the ordinary use of "server" and "client" in other contexts. For basics refer to `X(7)`, the LDP XWindow-User-HOWTO (`http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html`), and Remote X Apps mini-HOWTO (`http://www.tldp.org/HOWTO/mini/Remote-X-Apps.html`).

There are several ways of getting the "X server" (display side) to accept remote connections from an "X client" (application side):

- `xhost`

  - the host list mechanism (unsecure).
  - non-encrypted protocol (prone for eavesdropping attack).
  - Do not use this, if possible.
  - See 'Remote X connection: `xhost`' en la página 99 and `xhost(1x)`.

- `xauth`
  - the MIT magic cookie mechanism (unsecure but better than xhost).
  - non-encrypted protocol (prone for eavesdropping attack).
  - use this only for local connection.
  - See 'Gain root in X' en la página 100 and `xauth(1x)`.

- `xdm`, `wdm`, `gdm`, `kdm`, . . .
  - See `xdm(1x)` and `Xsecurity(7)` for the basics of X display access control.
  - See `wdm(1x)`, `gdm(8)`, and `kdm.options(5)` for more information, if these are installed.
  - See 'System-V `init` and runlevels' en la página 64 for how to disable `xdm` to gain Linux console upon boot without purging `xdm` package.

- `ssh -X`
  - port forwarding mechanism through secure shell (secure).
  - encrypted protocol (resource waste if used locally).
  - use this for remote connection.
  - See 'Remote X connection: `ssh`' en la página 99.

All remote connection methods, except `ssh`, require TCP/IP connection enabled on the X server. See 'TCP/IP connection to X' en la página siguiente.

9.4.1. X server

See `XFree86(1x)` for X server information.

To (re)configure X4 in woody, run:

```
# dpkg-reconfigure --p=low xserver-xfree86
```

Invoke X server from local console:

```
$ startx -- :<display> vtXX
eg.:
$ startx -- :1 vt8
... start on vt8 connected to localhost:1
```

9.4.2. X client

Most X client programs can be started as:

```
client $ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &
```

Here, these optional command line arguments mean:

- `-geometry` *`WIDTH`*`x`*`HEIGHT`*`+`*`XOFF`*`+`*`YOFF`*: the initial size and location of the window.
- `-fn` *`FONTNAME`*: the font to use for displaying text. *`FONTNAME`* can be:
  - a14: Normal size font
  - a24: Large size font
  - . . . (check available fonts by `xlsfont`.)
- `-display` *`displayname`*: the name of the X server to use. *`displayname`* can be:
  - *`hostname`*`:`*`D.S`* means screen *`S`* on display *`D`* of host *`hostname`*; the X server for this display is listening to TCP port 6000+D.
  - *`host`*`/unix:`*`D.S`* means screen *`S`* on display *D* of host *`host`*; the X server for this display is listening to UNIX domain socket `/tmp/.X11-unix/XD` (so it's only reachable from *`host`*).
  - *`:D.S`* is equivalent to *`host`*`/unix:`*`D.S`*, where *`host`* is the local hostname.

The default *`displayname`* for the X client program (application side) can be set by the DISPLAY environment variable. For example, prior to running an X client program, executing one of the following commands achieves this:

```
$ export DISPLAY=:0
        # The default, local machine using the first X screen
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

9.4.3. TCP/IP connection to X

Because remote TCP/IP socket connection without encryption is prone to the eavesdropping attack, the default setting for X of the recent Debian disables TCP/IP socket. Think about using `ssh` for remote X connection (see 'Remote X connection: `ssh`' en la página siguiente).

Method described here is not encouraged unless one is in a very secure environment behind a good firewall system with only trusted users present. Following configuration on the X server restores TCP/IP socket connection:

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Remove `-nolisten` to allow TCP/IP listening on the X server.

9.4.4. Remote X connection: `xhost`

`xhost` allows access based on hostnames. This is very insecure. The following will disable host checking and allow connections from anywhere if TCP/IP socket connection is allowed (see 'TCP/IP connection to X' en la página anterior):

```
$ xhost +
```

You can re-enable host checking with:

```
$ xhost -
```

`xhost` does not distinguish between different users on the remote host. Also, hostnames (addresses actually) can be spoofed.

This method must be avoided even with more restrictive host criteria if you're on an untrusted network (for instance with dialup PPP access to the Internet). See `xhost(1x)`.

9.4.5. Remote X connection: `ssh`

The use of `ssh` enables a secure connection from a local X server to a remote application server.

- Set `X11Forwarding` and `AllowTcpForwarding` entries to `yes` in `/etc/ssh/sshd_config` of the remote host.
- Start the X server on the local host.
- Open an `xterm` in the local host.
- Run `ssh` to establish a connection with the remote site.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
.....
```

- Run X application commands on the remote site.

```
loginname @ remotehost $ gimp &
```

This method allows the display of the remote X client output as if they were locally connected through local UNIX domain socket.

9.4.6. `xterm`

Learn everything about `xterm` at `http://dickey.his.com/xterm/xterm.faq.html`

9.4.7. Gain root in X

If a GUI program needs to be run with root privilege, use the following procedures to display program output on a user's X server. Never attempt to start an X server directly from the root account in order to prevent possible security risks.

Start X server as a normal user and open `xterm` console. Then:

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:*****
# printtool &
```

When using this trick to `su` to a non-root user, make sure `$HOME/.Xauthority` is group readable by this non-root user.

This command sequence can be automated by adding few files. From the root account, create the file `/etc/X11/Xsession.d/00xfree86-common_environment`, containing following lines:

```
if [ -f "$HOME/.xenvironment" ]; then
    . $HOME/.xenvironment
fi
```

From the user account, create the file `$HOME/.xenvironment` in his home directory, containing following lines:

```
# This makes X work when I su to the root account.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
```

Then run `su` (not `su -`) in an `xterm` window of the user. Now GUI programs started from this `xterm` can display output on this user's X window while running with root privilege. This trick works as long as default Xsession is executed. If a user set up his customization using `$HOME/.xinit` or `$HOME/.xsession`, above mentioned environment `XAUTHORITY` variable needs to be set similary in those scripts.

Alternatively, `sudo` can be used to automate the command sequence:

```
$ sudo xterm
... or
$ sudo -H -s
```

Here `/root/.bashrc` should contain:

```
if [ $SUDO_USER ]; then
    sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

This works fine even with the home directory of the user on an NFS mount, because root does not read the `.Xauthority` file.

There are also several specialized packages for this purpose: `kdesu`, `gksu`, `gksudo`, `gnome-sudo`, and `xsu`. Some other methods can be used to achieve similar results: creating a symlink from `/root/.Xauthority` to the user's corresponding one; use of the script sux (`http://fgouget.free.fr/sux/sux-readme.shtml`); or putting "`xauth merge ~`*USER_RUNNING_X*`/.Xauthority`" in the root initialization script.

See more on the debian-devel mailing list (`http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html`).

9.4.8. TrueType fonts in X

The standard `xfs` in XFree86-4 works fine with TrueType fonts. You have to install a 3rd party font server such as `xfs-xtt`, if you are using XFree86-3.

You just need to make sure that whatever apps you want to use the TrueType fonts are linked against libXft or libfreetype (you probably don't even have to worry about this if you're using precompiled .debs).

Remember to install required font files and generate the `fonts.{scale,dir}` files so that the fonts can be indexed and used.

Since **Free** fonts are sometimes limitted, installing or sharing some commercial TrueType fonts are option for a Debian users. In order to make this process easy fot the user, there have been some convienience packages:

- `ttf-commercial`
- `msttcorefonts` (No more useful as of 8/2002 due to M$ policy change)

You'll have a really good selection of TT fonts at the expense of contaminating your **Free** system with non-Free fonts.

9.4.9. Web Browser (graphical)

There are few web browser packages with graphical display capabilities available after woody release:

- `mozilla` The browser (new)
- `galeon` The browser with gnome UI (new)
- `konqueror` KDE browser
- `amaya-gtk` W3C reference browser
- `amaya-lesstif` W3C reference browser
- `netscape-...` (many, old)
- `communicator-...` (many, old)
- ...

The version of `mozilla` must match with the version `galeon` requires. Except UI, these 2 programs share HTML rendering engine (Gocko).

Plug in for the browsers such as `mozilla` and `galeon` can be enabled by installing "`*.so`" manually to the plugin directory and restart the browsers.

Plugin resources:

- Java plugin: install binary "J2SE" from `http://java.sun.com`.
- Flash plugin: install binary "Macromedia Flash Player 5" from `http://www.macromedia.com/software/flashplayer/`.
- `freewrl`: vrml browser and netscape plugin
- ...

9.4.10. CJK and X

Reference:

- 'Localization' en la página 109
- Suse pages for CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`)

Here, let us try for Japanese:

- install Japanese packages:
  - `kinput2-canna-wnn` — An input server for X11 applications.
  - `kterm`, `mlterm`, and `jfbterm`: Japanese compatible terminals.
  - `egg` — Input Method Architecture for Emacsen
  - `canna` — A Japanese input system (server and dictionary).
  - `freewnn-jserver` — Network-extensible Kana-to-Kanji conversion system.
  - . . . and all the Japanese font packages.

  In reality, use `tasksel` or `aptitude` to select "Japanese environment" and avoid installing softwares which conflict with the normal system.

- add a locale that supports Japanese characters (e.g. ja_JP.UTF-8, see 'Localization' en la página 109).

- add following environment values in `~/.xenvironment` using the same trick used in 'Gain root in X' en la página 100.

  ```
  XMODIFIERS="@im=kinput2"
  LC_CTYPE=ja_JP.UTF-8 # some japanese locale
  ```

  (Or manually do so in `xterm` before starting an application.)

- activate XIM `kinput2` by adding `*inputMethod: kinput2` to your X resources file (Looks like debian takes care of this).

- Some applications (such as `mlterm`) also allow you to set up `*inputMethod:` information dynamically at run time (press Ctrl-MouseBotton-3 in mlterm).

If you started the application, you should press "**Shift+Space**" and a window should popup stating that you now may input Japanese characters.

9.5. SSH

SSH (Secure SHell) is the secure way to connect over the Internet. A free version of SSH called openSSH is available as `ssh` package on Debian.

9.5.1. Basics

First install openSSH server and client.

```
# apt-get update && apt-get install ssh
```

The non-US entry in the `/etc/apt/source.list` is required. `/etc/ssh/sshd_not_to_be_run` must not be present if one wish to run openSSH server.

SSH has 2 authentication protocols:

- SSH protocol version 1:
  - potato version only supports this protocol.
  - available authentication methods:
    - RSAAuthentication: RSA identity key based user authentication
    - RhostsAuthentication: .rhosts based host authentication (insecure, disabled)
    - RhostsRSAAuthentication: .rhosts authentication combined with RSA host key (disabled)
    - ChallengeResponseAuthentication: RSA challenge-response authentication
    - PasswordAuthentication: password based authentication
- SSH protocol version 2:
  - post-woody versions use this as primary protocol.
  - available authentication methods:
    - PubkeyAuthentication: public key based user authentication
    - HostbasedAuthentication: .rhosts or /etc/hosts.equiv authentication combined with public key client host authentication (disabled)
    - ChallengeResponseAuthentication: chllenge-response authentication
    - PasswordAuthentication: password based authentication

Be careful about these differences if you are migrating to woody or using non-Debian system.

See `/usr/share/doc/ssh/README.Debian.gz`, `ssh(1)`, `sshd(8)`, `ssh-agent(1)`, and `ssh-keygen(1)` for details.

Followings are the key configuration files:

- `/etc/ssh/ssh_config`: SSH client defaults. See `ssh(1)`. Notable entries are:
  - `Host`: Restricts the following declarations (up to the next Host keyword) to be only for those hosts that match one of the patterns given after the keyword.
  - `Protocol`: Specifies the SSH protocol versions. The default is “2,1”.
  - `PreferredAuthentications` Specifies the SSH2 client authentication method. The default is “hostbased,publickey,keyboard-interactive,password”.

  - `ForwardX11`: The default is disabled. This can be overiden by a command line option "`-X`".

- `/etc/ssh/sshd_config`: SSH server defaults. See `sshd(8)`. Notable entries are:
  - `ListenAddress`: Specifies the local addresses sshd should listen on. Multiple options are permitted.
  - `AllowTcpForwarding`: The default is disabled.
  - `X11Forwarding`: The default is disabled.

- `$HOME/.ssh/authorized_keys`: the lists of the default public keys that client used to connect to this account on this host. See `ssh-keygen(1)`

- `$HOME/.ssh/identity`: See `ssh-add(1)` and `ssh-agent(1)`.

The following will start an `ssh` connection from client.

```
$ ssh username@hostname.domain.ext
$ ssh -1 username@hostname.domain.ext # Force SSH version 1
```

For the user, `ssh` functions as a smarter and more secure `telnet` (will not bomb with ^]).

9.5.2. Port forwarding – for SMTP/POP3 tunneling

To establish a pipe to connect to port 25 of *remote-server* from port 4025 of localhost, and to port 110 of *remote-server* from port 4110 of localhost through `ssh`, execute on the local machine:

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 \
           username@remote-server
```

This is a secure way to make connections to SMTP/POP3 servers over the Internet. Set `AllowTcpForwarding` entry to `yes` in `/etc/ssh/sshd_config` of the remote host.

9.5.3. Connect with fewer passwords

One can avoid having to remember a password for each remote system by using RSAAuthentication (SSH1 protocol) or PubkeyAuthentication (SSH2 protocol).

On the remote system, set respective entries, "RSAAuthentication yes" or "PubkeyAuthentication yes", in `/etc/ssh/sshd_config`.

Then generate authentication keys locally and install its public key in the remote system:

```
$ ssh-keygen          # RSAAuthentication: RSA1 key for SSH1
$ cat .ssh/id_rsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication: RSA key for SSH2
$ cat .ssh/id_rsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication: DSA key for SSH2
$ cat .ssh/id_dsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
```

One can change the passphrase later with “`ssh-keygen -p`”. Make sure to verify settings by testing the connection. In case of any problem, use “`ssh -v`”.

You can add options to the entries in `authorized_keys` to limit hosts and to run specific commands. See `sshd(8)` for details.

Note that SSH2 has `HostbaseAuthentication`. For this to work, you must adjust settings of `HostbasedAuthentication` to `yes` in both `/etc/ssh/sshd_config` on the server machine and `/etc/ssh/ssh_config` or `$HOME/.ssh/config` on the client machine.

9.5.4. Foreign SSH client

There are few free SSH client available on non-Unix like environment.

Windows puTTY (`http://www.chiak.greenend.org.uk/~sgtatham/putty/`) (GPL)

Windows (cygwin) SSH in cygwin (`http://www.cygwin.com/`) (GPL)

Macintosh macSSH (`http://www.macssh.com/`) (shareware?, GPL?)

See SorceForge.net, site documentation (`http://www.sourceforge.net/docman/?group_id=1`), “6. CVS Instructions”.

9.5.5. SSH agent

Just put your public key into `~/.ssh/authorized_keys`, and you’re all set:

```
$ ssh-agent
$ # paste the output to your shell
```

```
$ ssh-add .ssh/identity
$ # or ssh-add .ssh/id_dsa or whatever your private key is named
$ scp remote.host.with.public.key
```

For more, read `ssh-agent(1)` and `ssh-add(1)`.

9.5.6. Troubles

If you have problems, check the permissions of configuration files and run `ssh` with "`-v`" option.

Use "`-P`" option if you are root and have trouble with a firewall; this avoids the use of server ports 1–1023.

If `ssh` connections to a remote site suddenly stop working, it may be the result of tinkering by the sysadmin, most likely a change in `host_key` during system maintenance. After making sure this is the case and nobody is trying to fake the remote host by some clever hack, one can regain connection by removing the `host_key` entry from `$HOME/.ssh/known_hosts` on the local machine.

9.6. Mail programs

Mail configuration divides into three categories:

- MTA: `exim`
- MUA: `mutt`
- Utilities: `procmail, fetchmail, mail, ...`

9.6.1. Mail transport agent (Exim)

References:

- `exim-doc` and `exim-doc-html` packages
- `http://www.exim.org/`

Use `exim` as the mail transfer agent (MTA). Configure:

```
/etc/exim/exim.conf      "eximconfig" to create and edit
/etc/inetd.conf          comment out smtp to run exim as daemon
/etc/email-addresses     Add spoofed source address lists
check filters using exim -brw, -bf, -bF, -bV, ... etc.
```

A catchall for nonexistent email addresses (Exim)

In `/etc/exim/exim.conf` (Woody or later), in the DIRECTORS part, at the end (after the localuser: director) add a catch-all director that matches all addresses that the previous directors couldn't resolve (per Miquel van Smoorenburg):

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

If one wants to have more a detailed recipe for each virtual domain, etc., add the following at the end of `/etc/exim/exim.conf` (per me, not well tested):

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Then have an "*" entry in `/etc/email-addresses`

9.6.2. Mail utility (Fetchmail)

`fetchmail` is run in daemon mode to fetch mail from a POP3 account with an ISP into the local mail system. Configure:

```
/etc/init.d/fetchmail
/etc/rc?.d/???fetchmail run update-rc.d fetchmail default priority 30
/etc/fetchmailrc        configuration file (chown 600, owned by fetchmail)
```

Information on how to start fetchmail as a daemon from the `init.d` script for potato is confusing (woody fixed this). See /etc/init.d/fetchmail and /etc/fetchmailrc (`examples/`)

If your email headers are contaminated by ^M due to your ISP's mailer, add "stripcr" to your options in `$HOME/.fetchmailrc`:

```
options fetchall no keep stripcr
```

9.6.3. Mail utility (Procmail)

`procmail` is a local mail delivery and filter program. One needs to create `$HOME/.procmailrc` for each account that uses it. Example: _procmailrc (`examples/`)

9.6.4. Mail user agent (Mutt)

Use `mutt` as the mail user agent (MUA) in combination with `vim`. Customize with `~/.muttrc`; for example:

```
# use visual mode and "gq" to reformat quotes
set editor="vim -c 'set tw=72 et ft=mail'"
#
# header weeding taken from the manual (Sven's Draconian header weeding)
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

Add the following to `/etc/mailcap` or `$HOME/.mailcap` to display HTML mail and MS Word attachments inline:

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

9.7. Localization

Although Debian system is internationalized and supports conventions which depend on the language and the cultural rules (see 'Internationalization' en la página 24). User still needs to customize his working environment to his needs. This process is called localization.

9.7.1. Locales

A locale is a set of language and cultural rules. These cover aspects such as the language of system messages, different character sets, spelling conventions, etc. A program needs to be able to determine its locale and act accordingly to be portable to different cultures. Glibc offers support to this functionality to the program as a library. See `locale(7)`.

9.7.2. Activate locale support capability

Debian does **not** come with all available locales pre-compiled. Check `/usr/lib/locale` to see which locales (besides the default “C”) are compiled for your system. If the one you need is not present, you have two options:

- Edit `/etc/locale.gen` to add the desired locale, then run `locale-gen` as root to compile it. See `locale-gen(8)` and the manpages listed in its “SEE ALSO” section.
- Run `dpkg-reconfigure locales` to reconfigure the `locales` package.

9.7.3. Activate a particular `locale`

The following environment variables are evaluated in this order to provide particular locale values to programs:

1. LANGUAGE: This environment variable consists of a colon-separated list of locale names in order of priority. Used only if the POSIX locale is set to a value other than “C” [in woody; the potato version always has priority over the POSIX locale]. (GNU extension)

2. LC_ALL: If this is non-null, the value is used for all locale categories. (POSIX.1) Usually “” (null).

3. LC_*: If this is non-null, the value is used for the corresponding category (POSIX.1). Usually “C”.

   LC_* variables are:
   - LC_CTYPE: Character classification and case conversion.
   - LC_COLLATE: Collation order.
   - LC_TIME: Date and time formats.
   - LC_NUMERIC: Non-monetary numeric formats.
   - LC_MONETARY: Monetary formats.
   - LC_MESSAGES: Formats of informative and diagnostic messages and interactive responses.
   - LC_PAPER: Paper size.
   - LC_NAME: Name formats.
   - LC_ADDRESS: Address formats and location information.
   - LC_TELEPHONE: Telephone number formats.
   - LC_MEASUREMENT: Measurement units (Metric or Other).
   - LC_IDENTIFICATION: Metadata about the locale information.

4. LANG: If this is non-null and LC_ALL is undefined, the value is used for all LC_* locale categories with undefined values. (POSIX.1) Usually “C”.

Note that some applications (e.g. Netscape 4) ignore LC_* settings.

For example, in the case of the Italian-language manual pages, LC_MESSAGES needs to be set to `it`. The `man` program will then search for Italian manual pages under `/usr/share/man/it/`.

9.7.4. Beyond `locale`

Programs may need to be configured beyond `locale` configuration to enable comfortable working environment.

`language-env` package and its command `set-language-env` is a great helper script on Debian system.

Also language specific entries in “task” system accessed through `tasksel` or `aptitudes` is another useful resource.

For X example, see ‘CJK and X’ en la página 102.

For more information, see Suse pages for CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`). Also see the internationalization document, Introduction to i18n (`http://www.debian.org/doc/manuals/intro-i18n/`). It is aimed at developers but is also useful for system administrators.

Capítulo 10

Building a gateway with a Debian system

Debian offers an all-purpose gateway machine, which handles NAT, mail, DHCP, DNS cache, http proxy cache, CVS, NFS, and Samba services for a home LAN system.

10.1. Network configuration

10.1.1. Host configuration for the gateway

The LAN uses IP addresses for the following private network range to avoid IP address collision with the Internet.

```
Class A: 10.0.0.0                    with mask 255.0.0.0
Class B: 172.16.0.0 - 172.31.0.0     with mask 255.255.0.0
Class C: 192.168.0.0 - 192.168.255.0 with mask 255.255.255.0
```

Debian uses `/etc/network/interfaces` for IP configuration.

For example, if `eth0` connects to the Internet with a DHCP-provided IP address and `eth1` connects to the LAN, `/etc/network/interfaces` is set as following (Woody or later):

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.1
```

```
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
```

Issue the following command to update the networking configuration to the new `/etc/network/interfaces`:

```
# /etc/init.d/networking restart
```

Reminder: The `interfaces` file in Woody or later releases is not compatible with Potato.

If the system uses a PCMCIA NIC, one needs to set up the network through `/etc/pcmcia/network.opts` instead.

Check the output of the following if in doubt:

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg|more
```

Sometimes, DSL (PPPoE?) has MTU issues. Refer to the LDP DSL-HOWTO (`http://www.tldp.org/HOWTO/DSL-HOWTO/`).

10.1.2. IP-masquerade

Machines on the LAN can access Internet resources through a gateway which runs IP-masquerade (NAT).

```
# apt-get install ipmasq
```

Apply example rules to strengthen the `ipmasq` protection. See `/usr/share/doc/ipmasq/examples/stronger/README`. For Debian kernel-image-2.4, make sure to load the proper modules. See 'Network function' en la página 68.

For Debian kernel-image-2.2, edit `Z92timeouts.rul` in `/etc/masq/rules` as follows:

```
# tcp, tcp-fin, udp
# 2hr, 10 sec, 160 sec - default
# 1 day, 10 min, 10 min - longer example
$IPCHAINS -M -S 86400 600 600
```

Also, if the network is accessed through a PCMCIA NIC, `ipmasq` needs to be started from `/etc/pcmcia/network.opts`. Read `/usr/share/doc/ipmasq/ipmasq.txt.gz`.

10.1.3. Network configuration checkpoints

Typical set of programs:

```
# apt-get install nfs samba dhcpd dhcp-client bind squid procmail fetchmail
# apt-get install ssh cvs
```

Then check the following files:

```
/etc/init.d/dhcpd       (edit to serve only LAN = eth1)
/etc/host.allow         (ALL: 192.168.0.0/16 127.0.0.0/8) for NFS
/etc/exports            (Need this for NFS)
/etc/bind/db.192.168.1  (add)
/etc/bind/db.lan        (add)
/etc/bind/named.conf    (edit)
/etc/resolve.conf       (edit)
/etc/hosts
/etc/dhcpd.conf         (edit for LAN = eth1)
/etc/dhclient.conf      (edit to force local DNS)
/etc/samba/smb.conf
/etc/exim/exim.conf
/etc/mailname
/etc/aliases
/etc/squid.conf         (add all LAN host IP as allowed)
```

`bind` creates a local cache DNS server and changes DNS to localhost. Check `/etc/resolve.conf`:

```
nameserver 127.0.0.1
search lan.aokiconsulting.com
```

10.2. Manage multiple net connections

[FIXME] Policy routing (by Phil Brutsche pbrutsch@tux.creighton.edu): See the iproute manual (`http://lartc.org/`) for details. Traffic control (tc) may also be interesting.

Environment:

```
eth0: 192.168.1.2/24; gateway 192.168.1.1
eth1: 10.0.0.2/24; gateway 10.0.0.1
No masquerading on this machine.
```

Special magic:

```
# ip rule add from 192.168.1.2 lookup 1
# ip rule add from 10.0.0.2 lookup 2
# ip route add to default via 10.0.0.1 metric 0
# ip route add to default via 192.168.1.1 metric 1
# ip route add table 1 to 192.168.1.0/24 via eth0
# ip route add table 1 to 10.0.0.2/24 via eth1
# ip route add table 1 to default via 192.168.1.1
# ip route add table 2 to 192.168.1.0/24 via eth0
# ip route add table 2 to 10.0.0.2/24 via eth1
# ip route add table 2 to default via 10.0.0.2
```

[FIXME] I've never done this. How to setup dialup as backup to a fast connection?

Capítulo 11

Editores

11.1. Editores

Linux ofrece muchas alternativas en cuanto a editores de texto. Citemos algunos:

- vim: poderoso y pequeño editor patrimonio del BSD. Del inglés, VI iMproved.
- emacs: el editor definitivo patrimonio del GNU. Original de RMS (Richard M. Stallman).
- mcedit: editor GNU para principiantes. Idéntico al editor interno del mc.
- ae: pequeño editor que trae por defecto Potato. Evítelo.
- nano: pequeño editor GNU que trae por defecto Woody. Emula a pico.
- joe: para los nostálgicos usuarios del WordStar o TurboPascal.
- jed: editor rápido, con múltiples funciones, basado en menúes y con las combinaciones de teclas de emacs.
- jove: editor muy pequeño con las combinaciones de teclas de emacs.
- nvi: el nuevo vi. Totalmente compatible con el vi original.

Utilice el comando `update-alternatives --config editor` para cambiar el editor por defecto.

11.2. Emacs y Vim

11.2.1. Comandos útiles en Vim

```
<F1>            Ayuda
<esc>           Retorno al modo normal
V               Modo visual
i               Modo edición
:               Modo línea de comandos
```

:set tw=72	Establecer el ancho de texto en 72
<F11>	Modo edición (pegar)
:r! date -R	Inserta la fecha según la RFC-822
q*a*	Grabar lo que se escribe en el registro *a*
@*a*	Ejecutar lo que se escribió en el registro *a*

`q` y `@` pueden utilizarse para grabar y reproducir macros sencillas. Por ejemplo, para crear una macro que inserte etiquetas HTML para cursiva alrededor de una palabra escriba `qii<i>^[ea</i>^[q` (donde `^[` es la tecla ESC). Entonces, al escribir `@i` al inicio de una palabra se le añadirán la etiquetas <i> y </i>.

11.2.2. Comandos útiles en Emacs

<F1>	Ayuda
<F10>	Menú
C-u M-! date -R	Inserta la fecha RFC-822

11.2.3. Ejecutando el editor

Iniciar el editor:	emacs nom_archivo	vim nom_archivo
Iniciar el modo compatible con vi:		vim -C
Iniciar el modo no compatible con vi:		vim -N
Iniciar con el compilador por defecto:	emacs -q	vim -N -u NONE

11.2.4. Resumen de los comandos del editor (Emacs, Vim)

Salir:	C-x C-c	:qa /:wq /:xa /:q!
Volver al modo comando:	C-g	<esc>
Retroceder (flecha izquierda):	C-b	h
Avanzar (flecha derecha):	C-f	l
Siguiente (flecha hacia abajo):	C-n	j
Anterior (flecha hacia arriba):	C-p	k
Comienzo de línea (^):	C-a	0
Final de línea ($):	C-e	$
Comandos múltiples:	C-u nnn cmd	:count cmd
Comandos múltiples:	M-número cmd	
Guardar archivo:	C-x C-f	:w archivo
Comienzo del búfer:	M-<	1G
Final del búfer:	M->	G

Avanzar 1 pantalla:	C-v	^F
Avanzar 1/2 pantalla:		^D
Avanzar una 1 línea:		^E
Retroceder 1 pantalla:	M-v	^B
Retroceder 1/2 pantalla:		^U
Retroceder 1 línea:		^Y
Ir a la otra ventana:	M-C-v	
Borrar bajo el cursor:	C-d	x
Borrar desde el cursor al final de la línea:	C-k	D
Búsqueda incremental hacia adelante:		C-s
Búsqueda incremental hacia atrás:		C-r
Buscar hacia adelante:	C-s enter	/
Buscar hacia atrás:	C-r enter	?
Búsqueda incremental exp. reg:		M-C-s
Búsqueda incremental hacia atrás exp. reg:	M-x isearch-backward-regexp	
Buscar expresión regular:	M-C-s enter	/
Búsqueda hacia atrás exp. reg:	M-x isearch-backward-regexp enter	?
Ayuda:	C-h C-h	:help
Ayuda Apropos:	C-h a	
Ayuda combinaciones de teclas	C-h b	:help [tecla]
Ayuda Info:	C-h i	
Ayuda modo mayor:	C-h m	
Ayuda tutorial:	C-h t	:help howto
Deshacer:	C-	u
Rehacer:	C-f	^R
Marcar posición del cursor:	C-@	m{a-zA-Z}
Intercambiar marca y posición:	C-x C-x	
Ir a la marca del archivo actual:		’{a-z}
Ir a la marca en cualquier archivo:		’{A-Z}
Copiar región:	M-w	{visual}y
Borrar región:	C-w	{visual}d
Copiar y conservar en el búfer:	C-y	
Pegar desde el siguiente búfer:	M-y	p
Convertir una región en mayúsculas:	C-x C-u	{visual}U
Convertir una región en minúsculas:	C-x C-l	{visual}u
Insertar un carácter especial:	C-q numoctal/tecla	^V decimal/tecla
Reemplazar:	M-x replace-string	:%s/aaa/bbb/g
Reemplazar expreg:	M-x replace-regexp	:%s/aaa/bbb/g
Buscar y reemplazar:	M-%	:%s/aaa/bbb/gc
Buscar y reemplazar:	M-x query-replace	
Buscar y reemplazar exp. reg:	M-x query-replace-regexp	
Abrir archivo:	C-x C-f	:r archivo

```
Guardar archivo:                             C-x C-s             :w
Guardar todos los búferes:                   C-x s               :wa
Guardar como:                                C-x C-w archivo     :w archivo
Pedir un búfer:                              C-x b
Listar búferes:                              C-x C-b             :buffers
Cambiar a sólo lectura:                      C-x C-q             :set ro
Pedir y eliminar búfer:                      C-x k
División vertical:                           C-x 2               :split
División horizontal:                         C-x 3               :vsplit (ver. 6)
Moverse a otra ventana:                      C-x o               ^Wp
Eliminar esta ventana:                       C-x 0               :q
Eliminar otra(s) ventana(s):                 C-x 1               ^Wo
Ejecutar shell en bg:                        M-x compile
Matar shell en bg:                           M-x kill-compilation
Ejecutar make                                                    :make Makefile
Analizar mensaje de error:                   C-x`                :echo errmsg
Ejecutar shell y grabar:                     M-x shell           :!script -a tmp
...clean BS, ...                                                 :!col -b <tmp >r
...guardar/recordar grab. del shell:         C-x C-w record      :r record
Ejecutar shell:                              M-! sh              :sh
Ejecutar comando:                            M-! cmd             :!cmd
Ejecutar comando e insertar:                 C-u M-! cmd         :r!cmd
Ejecutar filtro:                             M-| archivo         {visual}:w archi
Ejecutar filtro e insertar:                  C-u M-| filter      {visual}:!filter
Mostrar opción                                                   :se[t] {option}?
Volver opción a valor predeterminado                             :se[t] {option}&
Resetea opción booleana                                          :se[t] no{option
Conmutar una opción booleana                                     :se[t] inv{optio
Ajustar el texto en 72 columnas                                  :se tw=72
No ajustar texto                                                 :se tw=0
Sangrado automático                                              :se ai
Expandir tabulación                                              :se et
Especificar comentario (correo)                                  :se comments=n:>

Ejecutar GDB                                 M-x gdb
Describe el modo GDB                         C-h m
Saltar una línea                             M-s
Siguiente línea                              M-n
Saltar una instrucción (stepi)               M-i
Finalizar el marco de la pila actual         C-c C-f
Continuar                                    M-c
up arg frames                                M-u
```

```
down arg frames                               M-d
Copiar número a partir del punto, insertar al final   C-x &
Colocar un punto de ruptura                   C-x SPC
```

11.2.5. Configuración de Vim

Para utilizar todas las características de Vim y el resaltado de sintáxis incluya las siguientes líneas en el `~/.vimrc` o `/etc/vimrc`:

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

El modo 'paste' permite evitar que el sangrado automático interfiera con las operaciones pegar/cortar en una terminal o consola. Es mejor que escribir simplemente ":set noai".

11.2.6. Ctags

`apt-get install exuberant-ctags` y ejecute ctags sobre los archivos fuente. En vim escriba `:tag` *`nombre_función`* para ir a la línea donde comienza *nombre_función*. Las marcas funcionan para C, C++, Java, Python y muchos otros lenguajes de programación.

Emacs utiliza las mismas ctags.

11.2.7. Convertir un porción de texto seleccionado en código HTML

`so \$VIMRUNTIME/syntax/2html.vim` desde el modo comando de Vim convertirá el texto seleccionado en código HTML. Guárdelo como `:w archivo.html` y luego escriba `:q`. Útil cuando se programa en C, etc.

11.2.8. Dividir la pantalla con `vim` puede editar múltiples archivos en un entorno de múltiples ventanas. Escriba ":help usr_08.txt" para más detalles.

Para dividir la pantalla para mostrar diferentes archivos, escriba en la línea de comandos de vi:

```
:split otro-archivo
:vsplit otro-archivo
```

o en la línea de comandos del shell:

```
$ vi -o archivo1.txt archivo.txt    # División horizontal
$ vi -O archivo1.txt archivo2.txt   # División vertical
```

en ambos casos se podrá trabajar con múltiples ventanas.

```
$ vimdiff archivo.txt~ archivo.txt
$ gvimdiff archivo.txt~ archivo.txt # en X
```

proporcionan una vista muy cómoda para comparar un archivo con su copia de respaldo. En SGML analiza las etiquetas y por lo tanto resulta muy útil cuando se trata de comparar traduciones.

Movimientos del cursor especiales con los comandos CTRL-W:

```
CTRL-W +      incrementar el tamaño de una ventana
CTRL-W -      disminuir el tamaño de una ventana
CTRL-W j      ir a la ventana inferior
CTRL-W k      ir a la ventana superior
...
```

Los siguientes comandos permiten controlar el desplazamiento de la pantalla:

```
:set scrollbind
:set noscrollbind
```

Capítulo 12

CVS

Consulte `/usr/share/doc/cvs/html-cvsclient`, `/usr/share/doc/cvs/html-info`, `/usr/share/doc/cvsbook` con la ayuda de lynx o ejecute `info cvs` o `man cvs` para una información más detallada.

12.1. Instalar el servidor CVS

La siguiente configuración permite que sólo un miembro del grupo "src" pueda hacer envíos al repositorio CVS (mediante el comando 'commit') y que la administración del CVS pueda llevarse a cabo únicamente por un miembro del grupo "staff" de modo de reducir las posibilidades de conflicto.

```
# cd /var/lib; umask 002 ; sudo mkdir cvs # [WOODY] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .         # "staff": restricción importante para un
                           #  proyecto nuevo
# chmod 3775 .             #  Si usa "staff", escribir 2775
# cvs -d /var/lib/cvs init #  ¡resulta más seguro especificar
                           #  explícitamente -d!
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

12.2. Sesiones CVS de ejemplo

12.2.1. CVS anónimo (únicamente para descargar)

```
$ export CVSROOT=:pserver:anonymous@cvs.qref.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

12.2.2. Uso del servidor CVS local

```
$ export CVSROOT=/var/lib/cvs
```

12.2.3. Uso del pserver en un CVS remoto

No resulta seguro pero es conveniente para un CVS anónimo:

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

12.2.4. Uso de un CVS remoto mediante `ssh`

Puede utilizar la autenticación RSA (‘Connect with fewer passwords’ en la página 105) que evita que se solicite la contraseña.

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

o para Sourceforge:

```
$ export CVSROOT=:ext:account@cvs.qref.sf.net:/cvsroot/qref
```

12.2.5. Crear un archivo CVS nuevo

Para,

```
ITEM                   VALOR                SIGNIFICADO
Árbol del proyecto:    ~/proyecto-x         Todos los archivos fuente
Nombre del proyecto:   proyecto-x           Nombre para este proyecto
```

```
Nombre del proveedor: Rama-principal      Etiqueta para toda la rama
Etiqueta de versión:  Versión-inicial     Etiqueta para una versión
                                          específica
```

Entonces,

```
$ cd ~/proyecto-x            # para entrar en el directorio del proyecto
 ... crear un árbol ...
$ cvs import -m "Comienzo del proyecto-x" proyecto-x Rama Versión
$ cd ..; rm -R ~/proyecto-x
```

12.2.6. Trabajando con CVS

Para recordar y trabajar con los fuentes locales del proyecto-x en un archivo CVS:

```
$ cd                           # para ubicarse en la zona de trabajo.
$ cvs co proyecto-x            # copia los fuentes del repositorio CVS
                               # a la máquina local
$ cd proyecto-x
 ... efectuar los cambios ...
$ cvs diff -u                  # similar a diff -u repository/ local/
$ cvs ci -m "Describir cambio"  # guarda los fuentes locales en el CVS
$ vi archivo_nuevo
$ cvs add archivo_nuevo
$ cvs ci -m "Se añadió archivo_nuevo"
$ cvs up                 # actualiza a la última versión del CVS
$ cvs tag Release-1      # añade etiqueta de versión
... hacer otras modificaciones ...
$ cvs tag -d Release-1   # elimina etiqueta de versión
$ cvs ci -m "otros comentarios"
$ cvs tag Release-1      # vuelve a añadir la etiqueta de versión
$ cd                     # para volver al área de trabajo.
$ cvs co -r Versión-inicial -d antiguo proyecto-x
... obtiene la versión original y la ubica en el directorio ’antiguo’
$ cd antiguo
$ cvs tag -b Versión-inicial-modif # crea la etiqueta de la rama (-b)
 ... Ahora puede trabajar sobre la versión original
$ cvs update -r Versión-inicial-modif
 ... En el árbol del proyecto ahora aparece la rama
     Versión-inicial-modif
```

```
 ... Trabaje en esta rama
$ cvs up              # sicronice los cambios en la rama con el resto
$ cvs ci -m "verificación hecha en la rama"
$ cvs update -r HEAD  # cambia rama al HEAD del tronco principal
$ cvs update -j Versión-inicial-modif
 ... se funde rama con el tronco principal
 ... resolver conflictos con el editor
$ cvs ci -m "Creación Versión-inicial-modif en el tronco principal"
$ cd
$ tar -cvzf antiguo-projecto-x.tar.gz antiguo  # crear archivo, -j para
                                               # bz2
$ cvs release -d antiguo           # eliminar fuentes locales (opcional)
```

Algunas opciones útiles para recordar (utilice a continuación del comando `cvs`):

```
-n      no ejecuta ningún comando que cambie el disco
-t      muestra los mensajes de la actividad en el cvs
```

12.2.7. Exportar archivos desde el CVS

Para obtener la última versión del CVS use "tomorrow":

```
$ cvs ex -D tomorrow nombre_módulo
```

12.2.8. Administrar el CVS

Añadir un alias al proyecto (servidor local):

```
$ su - admin           # un miembro del equipo
$ export CVSROOT=/var/lib/cvs ; cvs co CVSROOT/modules
$ cd CVSROOT
$ echo "px -a proyecto-x" >>modules
$ cvs ci -m "Ahora px es un alias del proyecto-x"
$ cvs release -d .
$ exit                 # control-D para salir de 'su'
$ cvs co -d proyecto px
 ... proyecto-x (alias:px) del CVS al directorio proyecto
$ cd proyecto
 ... realizar cambios ...
```

12.3. Resolución de problemas

12.3.1. Permisos de los archivos en el repositorio

CVS no sobreescribe el archivo del repositorio actual sino que lo reemplaza por otro. Por lo tanto el *permiso de escritura en el directorio del repositorio* resulta crítico. Cada vez que se cree un nuevo repositorio ejecute lo siguiente para evitar cualquier problema.

```
# cd /var/lib/cvs
# chown -R root:src repositorio
# chmod -R ug+rwX   repositorio
# chmod    2775     repositorio  # si es necesario para éste y sus
                                 # subdirectorios
```

12.3.2. El bit de ejecución

Cuando se abandona la sesión (check-out) se conserva el bit de ejecución. Ante cualquier problema de permisos de ejecución sobre un archivo puede cambiar sus permisos en el repositorio CVS con el siguiente comando.

```
# chmod ugo-x nombre_archivo
```

12.4. Comandos del CVS

Veamos los comandos del CVS con sus respectivos atajos.

```
{add|ad|new} [-k kflag] [-m 'mensaje'] archivos...
{admin|adm|rcs} [opciones-rcs] archivos...
{annotate|ann} [opciones] [archivos...]
{checkout|co|get} [opciones] módulos...
{commit|ci|com}   [-lnR]  [-m  'mensaje_registro'  |  -f  archivo] \
        [-r revision] [archivos...]
{diff|di|dif} [-kl] [opciones_rcsdiff] [[-r rev1 | -D fecha1] \
        [-r rev2 |  -D fecha2]] [archivos...]
{export|ex|exp} [-flNn] -r rev|-D fecha [-d dir] [-k kflag] módulo...
{history|hi|his} [-report] [-flags] [-options args] [archivos...]
{import|im|imp} [-options] repositorio nombre_proveedor etiq_versión...
{login|logon|lgn}
```

```
{log|lo|rlog} [-l] opciones-rlog [archivos...]
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] módulos...
{release|re|rel} [-d] directorios...
{remove|rm|delete} [-lR] [archivos...]
{rtag|rt|rfreeze} [-falnR]  [-b]  [-d]  [-r  etiqueta  |  -D  fecha] \
         sym_bolic_tag módulos...
{status|st|stat} [-lR] [-v] [archivos...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r etiqueta | -D fecha]  [-f] \
         sym_bolic_tag [archivos...]
{update|up|upd} [-AdflPpR] [-d] [-r tag|-D fecha] archivos...
```

Capítulo 13

Programación

No use la palabra “test” para designar un archivo ejecutable de prueba. `test` es un comando del shell.

13.1. Dónde empezar

Linux Programming Bible (John Goerzen/IDG books)

Documentos y ejemplos de `/usr/share/doc/<paquetes>`

Una gran cantidad de extensos documentos con información se puede obtener del proyecto GNU (`http://www.gnu.org/`).

Las siguientes 4 secciones contienen algunos scripts de ejemplo en diferentes lenguajes de programación para crear un archivo de texto con información de las cuentas para agregarse al archivo `/etc/passwd` usando un procesador por lotes como lo es el programa `newusers`. Cada script requiere como entrada de un archivo con una serie de líneas de la forma: `nombre apellido contraseña` (estos scripts no crearán los directorios ’home’ de los usuarios)

13.2. BASH

Learning the bash Shell, 2nd edition (O’Reilly)

```
$ info bash
$ mc   /usr/share/doc/bash/examples/ /usr/share/doc/bash/
```

(Instale el paquete `bash-doc` para ver archivos de ejemplo)

Pequeño programa de ejemplo (sirve como entrada para el comando `newusers`):

```
#!/bin/bash
# (C) Osmu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

13.3. AWK

Effective awk Programming, 3rd edition, and *sed & awk*, 2nd edition (O'Reilly)

```
$ man awk
$ info awk
```

Pequeño programa de ejemplo (sirve como entrada para el comando `newusers`)

```
#!/usr/bin/awk -f
# Script para crear un archivo que sirva para usar con el comando
# newusers a partir de un archivo que contiene las IDs y contraseñas de
# los usuarios de la siguiente manera: nombre apellido contraseña
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Distribuido bajo GNU GPL v 2 o cualquier versión posterior.
# Este programa se distribuye SIN NINGUNA GARANTÍA.

BEGIN {
    # Asignar UID, GID iniciales
    if ( ARGC > 2 ) {
 startuid = ARGV[1]
 delete ARGV[1]
    }
    else {
        printf( "Uso: newusers UIDinicial archivo\n" \
   "...donde UIDinicial es el ID del primer usuario " \
```

```
      "a agregar y 'archivo' es \n" \
      "un archivo de entrada de la forma: \n"\
      "'nombre apellido contraseña'\n" \
 )
 exit
    }

    infile = ARGV[1]
    printf( "Primer UID: %s\n\n", startuid )
}

/^#/ { next }

{
    ++record
    first = $1
    last = $2
    passwd = $3
    user= substr( tolower( first ), 1, 1 ) tolower( last )
    uid = startuid + record - 1
    gid = uid
    printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
 user, passwd, uid, gid, first, last, user \
 )
}
```

13.4. PERL

Programming Perl, 3rd edition (O'Reilly)

```
$ man perl
```

Pequeño programa de ejemplo (sirve como entrada para el comando `newusers`)

```
#!/usr/bin/perl
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
 if (/^#/) { next;}
        chop;
```

```
        $pid++;
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                  ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Instalar el módulo Perl *nombre módulo*:

```
# perl -MCPAN -e 'install nombre_modulo'
```

13.5. PYTHON

Learning Python (O'Reilly). Es un intérprete muy agradable.

```
$ man python
```

Pequeño programa de ejemplo (sirve como entrada para el comando `newusers`)

```
#! /usr/bin/env python
import sys, string

# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Adaptado del script awk por KMSelf Sat Aug 25 20:47:38 PDT 2001
# Este programa se distribuye SIN NINGUNA GARANTÍA.

def usages():
    print \
"Uso:  ", sys.argv[0], " UID_inicial [nombre_archivo]\n" \
"\tUID_inicial es el ID del primer usuario a agregar.\n" \
"\tnombre_archivo es el nombre del archivo de entrada.\n" \
"Si no se especifica, se toma la entrada estándar.\n\n" \
"Formato del archivo de entrada:\n"\
"\tnombre apellido contraseña\n"
 return 1

def parsefile(startuid):
    #
    # filtrado principal
    #
```

```
    uid = startuid
    while 1:
        line = infile.readline()
        if not line:
            break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # lo anterior falla con un # equivocado de parámetros :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s%s,,/home/%s:/bin/bash\n"%  \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid

if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
        #print "# UID empieza desde:%d\n"% uid
        if len(sys.argv) > 1:
            infilename   = string.join(sys.argv[2:])
            infile = open(infilename, 'r')
            #print "# Leer archivo desde:%s\n\n"% infilename
        else:
            infile = sys.stdin
        parsefile(uid)
```

13.6. MAKE

Managing Projects with make, 2nd edition (O'Reilly)

```
$ info make
```

Variables automáticas sencillas:

Sintáxis de las reglas:

```
Objetivo: [ Prerequisito ... ]
```

```
< TAB > comando1
< TAB > -comando2 # ignorar errores
< TAB > @comando3 # evitar repetición
```

Cada línea es interpretada por el shell antes de la sustitución de variables por parte de make. Utilice la \ al final de la línea para continuar el script. Utilice `$$` para escribir el `$` para las variables de entorno para un script de shell.

Reglas implícitas equivalentes:

```
.c:   header.h == %  : %.c header.h
.o.c: header.h == %.c: %.o header.h
```

Variables automáticas para las reglas anteriores:

```
foo.o: nuevo1.c nuevo2.c.c original1.c nuevo3.c
$@ == foo.o                                  (objetivo)
$< == nuevo1.c                               (el primero)
$? == nuevo1.c nuevo2.c nuevo3.c             (los nuevos)
$^ == nuevo1.c nuevo2.c original1.c nuevo3.c (todos)
$* == '%' patrón correspondiente al patrón objetivo.
```

Referencia de las variables:

```
foo1 := bar    # Expansión única
foo2  = bar    # Expansión recursiva
foo3 += bar    # Añade
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Extrae "directory"
$(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Ejecute `make -p -f/dev/null` para ver las reglas automáticas internas.

13.7. C

Kernighan & Ritchie, *The C Programming Language*, 2nd edition (Prentice Hall). Para consultar sobre las funciones de biblioteca de C en GNU:

```
# apt-get install glibc6-doc manpages-dev
```

Para consultar sobre las funciones de biblioteca de C, escriba “`info libc`”. Cada función C está documentada en la sección 3 del manual del sistema. Ejemplo: `man 3 printf`.

13.7.1. Programa sencillo en C (gcc)

Un simple ejemplo `example.c` para compilar con la biblioteca `libm` y obtener el ejecutable `run_example`:

```
$ cat > exemple.c
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
    double x;
    char y[11];
    x=sqrt(argc+7.5);
    strncpy(y, argv[0], 10); /* evita el desbordamiento del búfer */
    y[10] = ’\0’; /* para asegurar que la cadena termine con un ’\0’ */
    printf("%5i,%5.3f,%10s,%10s\n", argc, x, y, argv[1]);
    return 0;
}

$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
    1, 2.915, ./run_exam,     (null)
$ ./run_exam, 1234567890qwerty
    2, 3.082, ./run_exam, 1234567890qwerty
```

Aquí, `-lm` se necesita para enlazar la biblioteca `libm` y poder así usar sqrt(). La biblioteca actual se encuentra en el directorio `/lib` con el nombre `libm.so.6` que es un enlace lógico a `libm-2.1.3.so`.

Observe el último parámetro del texto de salida. Existen más de 10 caracteres a pesar de haber especificado `%10s`.

El uso de funciones que efectuan operaciones con punteros sin verificar sus limítes, tales como `sprintf` y `strcpy`, es censurado pues no evita las vulnerabilidades que surgen por desbordamiento de búfers. Utilice, en cambio, `snprintf` y `strncpy`.

13.7.2. Depurar (`gdb`)

GDB tutorial (http://www.dirac.org/linux/gdb/)

Utilice `gdb` para depurar un programa compilado con la opción `-g`. Muchos de los comandos se pueden abreviar. La expansión del tabulador funciona de igual manera que en el shell.

```
$ info gdb
...
$ gdb programa
(gdb) b 1                # coloca un punto de ruptura en la línea 1
(gdb) run arg1 arg2 arg3 # ejecuta programa
(gdb) next               # va a la siguiente línea
...
(gdb) step               # avanza un paso
...
(gdb) p parm             # imprime parámetro
...
(gdb) p parm=12          # inicializa el valor del parámetro en 12
```

Pueden también resultar útiles los siguientes comandos.

- `ldd`: muestra las dependencias de las bibliotecas compartidas
- `strace`: rastrea las señales y llamadas al sistema
- `ltrace`: rastrea las llamadas a las bibliotecas

Para depurar dentro de Emacs, diríjase a 'Resumen de los comandos del editor (Emacs, Vim)' en la página 118.

13.7.3. Flex – un Lex mejorado

Necesita proporcionar su propia `main()` y `yywrap()`, o su `programa.l` se vería así al compilar sin la biblioteca (`yywrap` es una macro; `%option main` activa en forma implícita a `%option noyywrap`):

```
%option main
%%
 .|\n ECHO ;
%%
```

Alternativamente, puede compilar con la opción `-lfl` del enlazador al final de la línea de comando de `cc` (como ATT-Lex con `-ll`). En este caso no se necesita la `%option`.

13.7.4. Bison – un Yacc mejorado

Necesita proporcionar su propia `main()` y `yyerror()`. `main()` llama a `yyparse()` que llama a `yylex()` que ha sido generalmente creada con FleX.

```
%%

%%    *

 * ltrace
```

13.7.5. Autoconf – desinstalar

SI cuenta con los fuentes, SI éstos utilizan autoconf/automake y SI puede recordar cómo lo configuró:

```
$ ./configure todas-las-opciones-de-configuración
# make uninstall
```

13.8. SGML

SGML permite la creación de múltiples formatos de un mismo documento. Un sistema SGML sencillo es Debiandoc que es el que se usó aquí. Este requiere de pequeñas adaptaciones en los archivos de texto originales para los siguientes caracteres:

```
<   &lt;
>   &gt;
&   &amp;
©   &copy;
-   &ndash;
--   &mdash;
```

Para marcar una sección como comentario, escriba:

```
<!-- El tema empieza aquí ... -->
```

Para marcar una sección que necesita modificarse, escriba:

```
<![%FIXME [ El tema empieza aquí ... ]]>
```

En SGML, la *primer definición* de una entidad gana. Por ejemplo:

```
<!entity% qref "INCLUDE">
<![%qref [ <!entity param "Datos 1"> ]]>
<!entity param "Datos 2">
&param;
```

Esto finaliza como "Datos 1". Si en la primer línea figurara "IGNORE" en vez de "INCLUDE" finalizaría como "Datos 2" (la segunda línea es una sentencia condicional).

Para más detalles: `apt-get install debiandoc-sgml-doc`. También, consulte a *DocBook: The Definitive Guide*, de Walsh y Muellner (O'Reilly)

13.9. Creación de paquetes Debian

Lea la documentación que viene con el paquete `packaging-manual` (Potato) o `debian-policy` (Woody).

Use `dh_make` del paquete `dh-make` para crear un paquete base. Luego, proceda según las instrucciones de `dh-make(1)` que utiliza `debhelper` en `debian/rules`.

Un enfoque alternativo consiste en usar `deb-make` del paquete `debmake`. No utiliza los scripts de `debhelper` y depende únicamente del shell.

Para ejemplos de paquetes con múltiples fuentes, véase "mc" (`dpkg-source -x mc_4.5.54.dsc`) que utiliza "sys-build.mk" de Adam Heath (`<doogie@debian.org>`) y "glibc" (`dpkg-source -x glibc_2.2.4-1.dsc`) que usa otro sistema creado por Joel Klecker (`<espy@debian.org>`).

Capítulo 14

GnuPG

Véase `/usr/share/doc/gnupg/README.gz` para una información más detallada o consulte `man gpg`.

14.1. Instalar Gnu PG

Lea el manual de GNU privacy (en Woody, `gnupg-doc`).

```
# gpg --gen-key                    # genera una clave nueva
# gpg --gen-revoke mi_usuario_ID   # genera una clave de revocación para
                                   # mi_usuario_ID
# host -l pgp.net | grep www|less # busca los servidores de claves pgp
```

Por el momento, buenos servidores de claves son:

```
keyserver wwwkeys.eu.pgp.net
keyserver wwwkeys.pgp.net
```

Debe tener cuidado de no crear más de dos subclaves. Si lo hace, los servidores de claves de pgp.net corromperán la clave. Asimismo, únicamente se debe especificar un servidor de claves en `$HOME/.gnupg/options`.

Desgraciadamente, los siguientes servidores ya no funcionan más:

```
keyserver search.keyserver.net
keyserver pgp.ai.mit.edu
```

14.2. Usar GnuPG

Manejo de archivos:

```
$ gpg  [opciones]  comando [args]
$ gpg {--armor|-a} {--sign|-s} archivo          # firma el 'archivo' en
                                                # el archivo.asc
$ gpg --clearsign archivo                       # firma el 'archivo'
                                                # sin cifrarlo
$ gpg --clearsign --not-dash-escaped patchfile  # firma patchfile sin
                                                # cifrar
$ gpg --verify archivo                          # verifica el 'archivo'
                                                # firmado sin cifrar
$ gpg -o archivo.firm {-b|--detach-sig} archivo # crea firma separada
$ gpg --verify archivo.firm archivo         # verifica el 'archivo'
                                            # con ayuda del archivo.firm
$ gpg -o archivo_cifrador {--recipient|-r} nombre {--encrypt|-e} archivo
        # cifrado con clave pública para el destinario nombre
$ gpg -o archivo_cifrador {--symmetric|-c} archivo # cifrado simétrico
$ gpg -o archivo --decrypt archivo_cifrador        # descifrado
```

14.3. Administrar GnuPG

Administración de claves:

```
$ gpg --edit-key ID_usuario       # "help" para ayuda interactiva
$ gpg -o archivo --exports        # exporta todas las claves al
                                  # 'archivo'
$ gpg --imports archivo           # importa todas las claves del
                                  # 'archivo'
$ gpg --send-keys ID_usuario      # envía la clave del ID_usuario al
                                  # servidor de claves
$ gpg --recv-keys ID_usuario      # recibe la clave del ID_usuario del
                                  # servidor de claves
$ gpg --list-keys ID_usuario      # lista las claves del ID_usuario
$ gpg --list-sigs ID_usuario      # lista las firmas del ID_usuario
$ gpg --check-sigs ID_usuario     # verifica la firma del ID_usuario
$ gpg --fingerprint ID_usuario    # verifica la huella dactilar del
                                  # ID_usuario
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
```

```
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys
              # obtiene claves desconocidas
              # actualiza las claves para todas las firmas desconocidas.
```

Códigos de confiabilidad:

```
-         No se asignó/evaluó confiabilidad del poseedor.
e         Ha fallado la evaluación de confiabilidad.
q         No existe suficiente información para realizar la evaluación.
n         No confiar nunca en esta clave.
m         Relativamente confiable.
f         Totalmente confiable.
u         Plenamente confiable.
```

14.4. Uso con Mutt

Agregar lo siguiente al `~/.muttrc` para evitar que GnuPG -un programa muy 'pesado'- arranque automáticamente y sólamente se active al pulsar la tecla 'S' en el menú del índice.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

Capítulo 15

Suporte para Debian

Es posible recurrir a las siguientes fuentes para obtener ayuda, consejo y soporte para Debian. En lo posible se deben usar estos recursos antes de ponerse a gritar en las listas de correo :)

Observése que puede acceder a una gran cantidad de documentación en su sistema proveniente de los distintos paquetes usando un navegador web o mediante los comandos ‘dwww’ o ‘dhelp’.

15.1. Referencias

Las siguientes referencias están disponibles para Debian y Linux en general. Si sus contenidos entran en conflicto, siempre confíe más en las fuentes de información primarias que en las fuentes de información secundarias tal como este documento.

- Manual de Instalación (primaria)
  - Leer antes de instalar o actualizar.
  - Web: `http://www.debian.org/releases/stable/installmanual`
  - Web: `http://www.debian.org/releases/testing/installmanual` (en preparación)
  - Paquete: `install-doc`
  - Archivo: `/usr/share/doc/install-doc/index.html`
- Nota de la versión (primaria)
  - Se debe leer antes de instalar o actualizar incluso si se piensa que que sabe absolutamente todo.
  - Web: `http://www.debian.org/releases/stable/releasenotes`

  - Web: http://www.debian.org/releases/testing/releasenotes (en preparación)
  - Paquete: `install-doc`
  - Archivo: `/usr/share/doc/install-doc/upgrade-i386.html[FIXME]`

- FAQ (secundaria)
  - Preguntas frecuentes (un tanto desactualizadas)
  - Web: http://www.debian.org/doc/manuals/debian-faq/
  - Paquete: `doc-debian`
  - Archivo: `/usr/share/doc/debian/FAQ/index.html`

- APT COMO (secundaria)
  - Detallada guía de usuario para la administración de paquetes Debian (woody)
  - Web: http://www.debian.org/doc/manuals/apt-howto/
  - Paquete: `apt-howto`,
  - Archivo: `/usr/share/doc/apt-howto`

- `dselect` Documentación para Principiantes (secundaria)
  - Tutorial de `dselect`
  - Web: http://www.debian.org/releases/woody/i386/dselect-beginner
  - Paquete: `install-doc`,
  - Archivo: `/usr/share/doc/install-doc/[FIXME]`

- Manual de Normativa de Debian (primaria)
  - Texto técnico de referencia de Debian.
  - Web: http://www.debian.org/doc/debian-policy/

- Referencia para los Desarrolladores de Debian (primaria)
  - Conocimientos básicos para desarrolladores.
  - Para consultar una vez para el resto de nosotros.
  - Web: http://www.debian.org/doc/manuals/developers-reference/

- Guía para Nuevos Encargados de Paquetes de Debian (primaria)
  - Guía práctica para desarrolladores.
  - Tutoriales para empaquetar para el resto de nosotros.

  - `http://www.debian.org/doc/manuals/maint-guide/`

- Manual para la creación de paquetes (potato)
  - Paquete `packaging-manual` de potato.
- Páginas del manual al estilo Unix (primaria)
  - `man <nombre-paquete>`
- Páginas info al estilo GNU (primaria)
  - `info <nombre-paquete>`
- Documentación específica de paquetes (primaria)
  - Encuéntrela en `/usr/share/doc/<nombre-paquete>`
- LDP: Proyecto de Documentación de Linux (secundaria)
  - COMOs y mini-COMOs para Linux
  - Web: `http://www.tldp.org/`
  - Paquete: `doc-linux`
  - Archivo: `/usr/share/doc/HOWTO/`
- DDP: Proyecto de Documentación de Debian (secundaria)
  - Manuales específicos de Debian
  - Web: `http://www.debian.org/doc/`
- El Rincón de los Desarrolladores de Debian (secundaria)
  - Información clave para desarrolladores de Debian
  - Muy instructiva para el usuario final
  - Web: `http://www.debian.org/devel/`
- Código fuente (absolutamente primaria)
  - Nadie puede negarlo :-)
  - Descargar el código fuente siguiendo 'The source code' en la página 11

15.2. Encontrar el significado de una palabra

Muchas palabras usadas en Debian son términos bastantes crípticos. También se utilizan muchos acrónimos. El siguiente comando resolverá la mayoría de nuestras dudas:

```
$ dict <escriba-aquí-una-palabra-extraña>
```

15.3. Sistema de seguimiento de fallos de Debian

La distribución Debian tiene un sistema de seguimiento de fallos (BTS) (`http://bugs.debian.org/`) que lleva un registro de fallos informados por los usuarios y desarrolladores. A cada fallo se le asigna un número y se mantiene en el archivo hasta que es marcado como resuelto.

Antes de enviar el informe de un fallo debe comprobar que nadie lo hizo antes. Las listas con los fallos más significativos están disponibles en Internet (`http://bugs.debian.org/`) y en otros lugares (`http://www.debian.org/Bugs/Access`). Véase también 'Verificar los bugs Debian' en la página 55.

La forma de reportar un fallo se explica en `http://www.debian.org/Bugs/Reporting`

15.4. Listas de Correo

Por lo menos lea debian-devel-announce (en inglés, de sólo lectura y con poco tráfico) para estar al día con Debian.

Las listas de correo de mayor interés para los usuarios de Debian son la debian-user (en inglés, de libre acceso y con mucho tráfico) y las otras listas debian-user-*idioma* (para otros idiomas).

Para información sobre estas listas y detalles de cómo suscribirse véase `http://lists.debian.org/`. Por favor, consulte los archivos tratando de encontrar respuestas a sus preguntas antes de publicarlas y ajústese a la reglas de etiqueta de la lista.

15.5. IRC

Debian tiene un canal IRC que proporciona soporte y ayuda a usuarios Debian en la red IRC de Proyectos Open. Esta red está dedicada a brindar información y compartir recursos para la comunidad Open Source. Para acceder al canal apunte su cliente favorito de IRC a irc.openprojects.net y únase a #debian.

Por favor, cumpla con las reglas del canal, respetando plenamente a los otros usuarios. Visite esta página (`http://www.openprojects.net`) para más información sobre los proyectos Open Source.

15.6. Motores de Búsqueda

Existen diversos motores de búsqueda que proporcionan documentación relacionada con Debian:

- Página de búsqueda de Debian en la WWW (`http://search.debian.org/`).
- Google (`http://www.google.com/`): incluir "site:debian.org" como parámetro de búsqueda.
- Grupos Google (`http://groups.google.com/`): un motor de búsqueda para los foros de discusión. Incluir "linux.debian.user" como parámetro de búsqueda.
- AltaVista (`http://www.altavista.com/`)

Por ejemplo, buscando la cadena "cgi-perl" se obtiene una explicación más detallada de este paquete que la breve descripción proporcionada por su archivo de control. Véase 'Verificar los bugs Debian' en la página 55 por ejemplo.

15.7. Páginas en Internet

Existen algunas páginas web con información de soporte general.

- Debian planet (`http://www.debianplanet.org/`)
- Linux.com (`http://linux.com/`)
- Las páginas de Matt Chapman (Guía Unix) (`http://www.belgarath.demon.co.uk/`)

Las siguientes son algunas URLs al azar que reuní sobre temas específicos.

- Adrian Bunk: paquetes para utilizar el kernel 2.4.x con Debian potato (`http://www.fs.tum.de/~bunk/kernel-24.html`)
- Linux en ordenadores portátiles (`http://www.linux-laptop.net/`)
- Xterm FAQ (`http://dickey.his.com/xterm/xterm.faq.html`)

- mini-COMO sobre el sistema de archivos EXT3 (`http://www.symonds.net/~rajesh/howto/ext3/index.html`)
- Soporte de archivos grandes en Linux (`http://www.suse.de/~aj/linux_lfs.html`)
- LNX-BBC (Business-card-sized boot CD project) (`http://www.lnx-bbc.org/`)
- Información sobre Linux de Karsten Self (Particiones, copias de respaldo, navegadores. . .) (`http://kmself.home.netcom.com/Linux/`)
- Copias de respaldo-COMO de Alvin Oga (`http://www.Linux-Backup.net/`)
- Seguridad-COMO de Alvin Oga (`http://www.Linux-Sec.net/`)
- Diversas fuentes NO OFICIALES para APT (`http://www.internatif.org/bortzmeyer/debian/apt-sources/`)
- Configuración de Ethernet en ordenadores portátiles (`http://www.orthogony.com/gjw/lap/lap-ether-intro.html`)

Apéndice A

Apéndice

A.1. Autores

La Guía de referencia Debian fue iniciada por Osamu Aoki <debian@aokiconsulting.com> a partir del resumen de su instalación personal que inicialmente denominó “Referencia Rápida. . . ” (“Quick Reference . . . ”). Muchos de los contenidos provienen de los archivos de la lista de correo debian-user. También fueron referenciados “Debian – Manual de Instalación” y “Debian – Release Notes”.

Siguiendo las sugerencias de Rodin, un miembro muy participativo del “Proyecto de Documentación Debian (DDP)” y el encargado actual de las “FAQ de Debian”, el documento fue renombrado como “Referencia Debian” y se nutrió con diversos capítulos de las “Debian FAQ”.

Este documento fue editado, traducido y ampliado por los siguientes miembros el equipo QREF:

- Inglés original de la “Quick Reference. . . ”
    - Osamu Aoki <debian@aokiconsulting.com>
- Relectura y reescritura al inglés
    - David Sewell <dsewell@virginia.edu>
    - Brian Nelson <nelson@bignachos.com>
    - Daniel Webb <webb@robust.colorado.edu>
- Traducción al francés
    - Guillaume Erbs <gerbs@free.fr>

    - Rénald Casagraude <rcasagraude@interfaces.fr>
    - Jean-Pierre Delange <delange@imaginet.fr>
    - Daniel Desages <daniel@desages.com>

- Traducción al italiano
    - Davide Di Lazzaro <mc0315@mclink.it>

QREF es una abreviatura del título del documento original, "Quick Reference. . . " (Referencia rápida. . .) y es también el nombre del proyecto en qref.sourceforge.net.

La mayoría de los contenidos de 'Debian fundamentals' en la página 5 provienen de las "FAQ de Debian" (Marzo del 2002):

- 5. Los archivos FTP de Debian "ftparchives.sgml" (todo el capítulo)
- 6. Fundamentos de la Administración de Paquetes Debian "pkg_basics.sgml" (todo el capítulo)
- 7. Las Herramientas de Administración de Paquetes Debian "pkgtools.sgml" (todo el capítulo)
- 8. Manteniendo su sistema Debian actualizado "uptodate.sgml" (todo el capítulo)
- 9. Debian y el kernel "kernel.sgml" (todo el capítulo)
- 10. Personalizando la instalación de Debian GNU/Linux "customizing.sgml" (parte del capítulo)

Estas secciones de las "FAQ de Debian" han sido incluidas en este documento después de algunas modificaciones para reflejar los cambios recientes del sistema Debian.

Las "FAQ de Debian" originales fueron creadas y estaban a cargo de J.H.M. Dassen (Ray) y Chuck Stickelman. Las personas que reescribieron las "FAQ de Debian" fueron Susan G. Kleinmann y Sven Rudolph. Posteriormente estuvieron a cargo de Santiago Vila. El encargado actual es Josip Rodin.

Parte de la información de las "FAQ de Debian" provienen de:

- El anuncio de la versión Debian-1.1, por Bruce Perens (`http://www.perens.com/`).
- Las FAQ de Linux, por Ian Jackson (`http://www.chiark.greenend.org.uk/~ijackson/`).
- Archivos de las Listas de Correo de Debian (`http://lists.debian.org/`),

- el manual de dpkg para programadores y el manual de normativa de Debian (véase ‘Referencias’ en la página 143)
- una gran cantidad de desarrolladores, voluntarios, beta-testers y
- los vagos recuerdos de sus autores :-)

Queremos agradecer a todos aquellos que han ayudado a crear este documento.

A.2. Garantías

Puesto que no soy un experto, no pretendo ser un entendido en Debian o Linux en general. Las consideraciones sobre seguridad quizás sean aplicables únicamente al usuario doméstico.

El presente documento no reemplaza a ninguna de las guías autorizadas.

No se ofrecen garantías de ningún tipo. Todas las marcas son propiedad de sus respectivos dueños.

A.3. Comentarios

Son bienvenidos todos los comentarios y sugerencias. Por favor, envíen un mensaje a Osamu Aoki (`http://www.aokiconsulting.com/`) <debian@aokiconsulting.com> en inglés o a cada traductor en su repectivo idioma.

Puesto que no soy un inglés nativo agradeceré cualquier corrección gramatical.

La mejor contribución consiste en aplicar diff a la versión SGML del documento aunque también es bienvenido diff aplicado a la versión texto.

A.4. Disponibilidad

La última versión oficial se puede obtener de `http://www.debian.org/doc/manuals/debian-reference/` y el proyecto se encuentra en `http://qref.sourceforge.net/`.

También se puede descargar en distintos formatos: texto plano, HTML, PDF y PostScript.

Los archivos originales en SGML usados para crear este documento están disponible vía CVS en: `:pserver:anonymous@cvs.qref.sf.net/cvsroot/qref` o en `http://qref.sourceforge.net/Debian/qref.tar.gz`.

A.5. Formato del documento

El presente documento fue escrito usando la DTD SGML de DebianDoc (re-escrito a partir del SGML de LinuxDoc). El sistema SGML de DebianDoc nos permite crear documentos en una amplia variedad de formatos a partir de un único archivo fuente. De esta manera es posible ver este documento en HTML, texto plano, TeX DVI, PostScript, PDF o GNU info.

Las utilidades de conversión del SGML de DebianDoc están disponible en el paquete Debian `debiandoc-sgml`.

A.6. El laberinto de Debian

El sistema Debian es una poderosa plataforma informática cuando es utilizado en red. No obstante, aprender a usar todas sus capacidades no es una tarea sencilla. La configuración de la impresora es un buen ejemplo.

Existe un mapa completo y detallado denominado "CÓDIGO FUENTE". Es muy preciso pero muy difícil de entender. Existen también referencias llamadas COMO (HOWTO) y mini-COMO (mini-HOWTO). Son más fácil de entender pero tienden a proporcionar demasiados detalles y a perder de vista la idea principal. A veces tengo problemas para encontrar la sección correcta en un extenso HOWTO cuando necesito sólo un par de comandos.

Para transitar por el laberinto de información de la configuración del sistema Linux, comencé escribiendo apuntes sencillos en archivos de texto como referencia rápida. La lista de archivos fue haciéndose más grande y mientras tanto aprendí debiandoc. El resultado es esta *Guía de referencia Debian*.

Tomé la decisión de no explicar lo que puede encontrarse en una página del manual, página info o en documentos HOWTO. Asimismo no intento explicar sino dar información directa mediante una secuencia exacta de comandos o guiones de ejemplo. La audiencia al cual va dirigida es la formada por aquellas personas capaces de leer los scripts de shell.

Mi principio rector es *Manténgalo breve y sencillo* (KISS con sus siglas en inglés).