Servidores Linux en centros educativos

22 de enero de 2005

Índice general

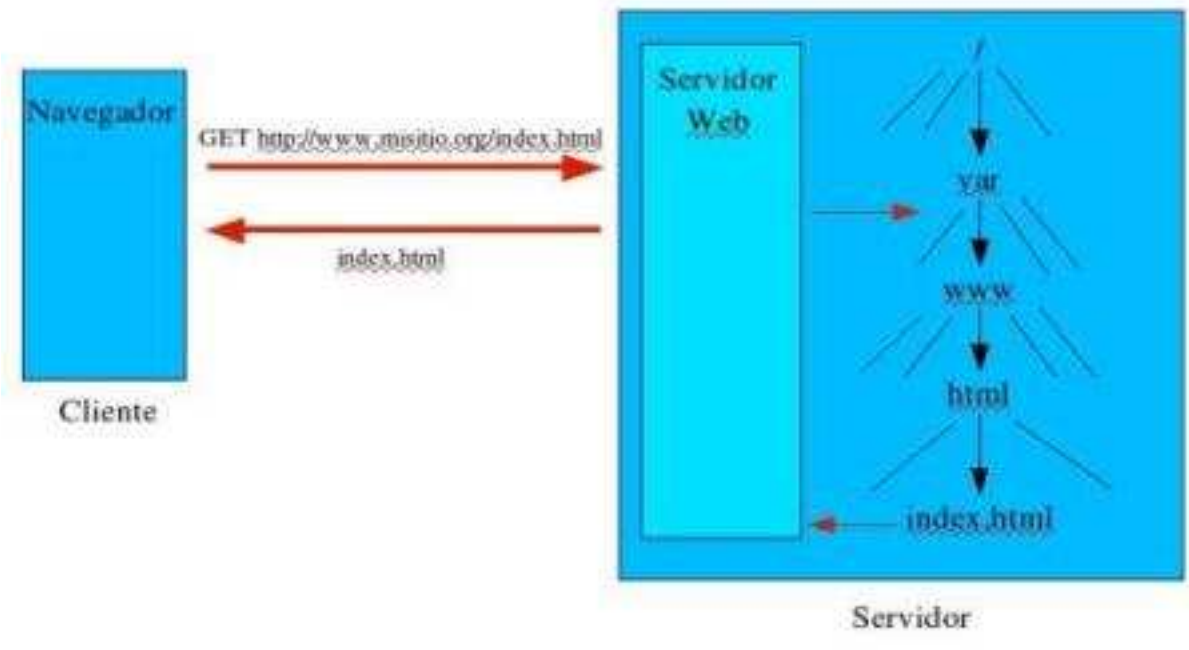
Servidores: Montaje y configuración de distintos servidores.

Capítulo 1

Servidor web. Apache.

1.1. Arquitectura web.

Para abrir una página Web en un navegador, normalmente se teclea la correspondiente URL o se pica en el hiperenlace oportuno. Una vez que se realiza esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de ficheros y la envía de vuelta al navegador que la solicitó.



1.1.1. El Navegador Web, Browser.

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles

errores que se puedan producir.

1.1.2. El Servidor Web.

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

1.1.3. HyperText Transfer Protocol, HTTP.

HTTP es un protocolo del nivel de aplicación para sistemas de información multimedia distribuidos. Es un protocolo no orientado a estado que puede ser utilizado para más propósitos que para manejar ficheros HTML. Para entendernos, son las normas que habrán de seguir cliente y servidor para comunicarse.

Entre las propiedades de HTTP se pueden destacar las siguientes:

- Utiliza un esquema de direccionamiento comprensible.

  Utiliza el Universal Resource Identifier (URI) para localizar sitios (URL) o nombres (URN) sobre los que hay que aplicar un método. La forma general de un URL es servicio://host/fichero.ext

- Arquitectura Cliente-Servidor.

  HTTP se asienta en el paradigma solicitud/respuesta.

- La comunicación se asienta sobre TCP/IP.

  El puerto por defecto es el 80, pero se pueden utilizar otros.

- Es un protocolo sin conexión y sin estado.

  Después de que el servidor ha respondido la petición del cliente, se rompe la conexión entre ambos. Además no se guarda memoria del contexto de la conexión para siguientes conexiones.

- Está abierto a nuevos tipos de datos.

  HTTP utiliza tipos MIME (Multipart Internet Mail Extension) para la determinación del tipo de los datos que transporta. Cuando un servidor HTTP transmite información de vuelta a un cliente, incluye una cabecera que le indica al cliente sobre los tipos de datos que componen el documento. De la gestión de esos datos se encargan las utilidades que tenga el cliente (visor de imágenes, de vídeo, etc.)

Una transacción HTTP está compuesta por una cabecera, y opcionalmente, por una línea en blanco seguida de los datos. En la cabecera se especifica tanto la acción solicitada en el servidor, como los tipos de datos devueltos o un código de estado.

1.1.4. La interacción entre el Navegador y el Servidor.

Durante una sesión normal de trabajo en WWW un cliente (navegador) solicita un documento de un servidor Web y una vez obtenido lo muestra al usuario que hizo la solicitud. Si este documento contiene un enlace a otro documento (en el mismo o en distinto servidor), y el usuario activa el enlace el cliente Web efectuará otra petición y mostrará el nuevo documento.

Durante la comunicación entre el cliente y el servidor HTTP en el que el cliente solicita el documento doc1.html al servidor se intercambian la siguiente transacción HTTP:

```
GET /doc1.html HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: jvegas@infor.uva.es
```

```
/* esto es una linea en blanco */
```

El método GET indica el fichero que el cliente solicita y la versión de HTTP. El cliente también muestra una lista de los tipos MIME que puede aceptar como retorno, además de identificar el navegador que utiliza (para que el servidor pueda optimizar los ficheros para el tipo particular de navegador) y su dirección de correo electrónico. Al final existe una línea en blanco que determina el final de la cabecera HTTP.

El servidor responde mandando la siguiente transacción HTTP:

```
HTTP/1.0 200 OK
Date: Friday, 23-Feb-01 16:30:00 GMT
Server: Apache/1.1.1
Content-type: text/html
Content-length: 230
/* esto es una linea en blanco */
<HTML><HEAD><TITLE>........ </HTML>
```

En este mensaje el servidor utiliza la versión 1.0 de HTTP, y manda el código de estado 200 para indicar que la petición del cliente ha sido procesada satisfactoriamente. También se identifica como un servidor Apache. Indica al cliente que el contenido del documento es texto en formato HTML y que tiene una longitud de 230 bytes.

1.1.4.1. Códigos de Respuesta del Servidor.

El servidor HTTP responde con un código que informa sobre el estado de la transacción. Los códigos se agrupan según las siguientes categorías:

Rango	Significado
100-199	Informativo
200-299	Éxito en la resolución de la petición
300-399	Petición redirigida, necesarias más acciones
400-499	Petición incompleta
500-599	Errores en el servidor

1.2. El servidor web Apache.

Poco después del nacimiento de la Web, un grupo del Centro Nacional de Actividades de Supercomputación (National Center for Supercomputing Activities), NCSA, de la Universidad de Illinois creó un servidor web (HTTPd NCSA) que fue el más usado en la web hasta mediados del año 1994.

Su principal desarrollador (Rob McCool) abandonó entonces NCSA y el proyecto. Sin embargo bastantes personas siguieron trabajando con HTTPd NCSA y así fueron surgiendo diversos parches para el código fuente.

Fue entonces cuando un grupo de desarrolladores (ocho miembros en principio) comenzaron a trabajar juntos sobre HTTPd y los parches que habían ido mejorandolo: surgía el proyecto Apache. La primera versión oficial, Apache 0.6.2, se lanzó en o abril de 1995 (el nombre proviene de "A PAtCHy" release, ya que en principio era una versión parcheada del HTTPd 1.3 NCo SA). El 1 de diciembre de 1995 se hizo publica la versión 1.0. Desde entonces Apache se ha convertido en el servidor web más usado (más del 60 % de todos los servidores).

En 1998 se llega a un acuerdo con IBM que permitiá conseguir que Apache funcionara también bajo Windows, convirtiéndose en una excelente alternativa a IIS (Microsoft Internet Information e Server). Apache es el servidor Web (protocolo HTTP) más utilizado en el mundo actualmente. Se encuentra muy por encima de sus competidores, ya sean gratuitos o comerciales. Por supuesto, es el más utilizado en sistemas Linux.

En su forma más simple, un servidor web transmite páginas en formato HTML a los navegadores cliente (Netscape, Explorer, Opera...). Pero el servidor web hoy día puede hacer mucho más, ya sea por sus propios medios o mediante su integración con otros programas. Existen varias formas en las que Apache puede proveernos contenidos:

- **Páginas estáticas**: Es el modo básico y más primitivo, pero que en un gran numero de casos es lo unico que se necesita: transferir ficheros HTML, imágenes... Puede que con un servidor Linux de bajas prestaciones (puede ser un 486) consigamos estupendos resultados, si es esto lo que queremos.
- **Contenido dinámico**: La información cambia constantemente y un medio para mantener nuestras páginas actualizadas es generarlas dinámicamente desde una base de datos, ficheros u otras fuentes de datos.

Apache posee muchas facilidades para generar este tipo de contenido.

1. **Soporte del protocolo HTTP 1.1**. Además mantiene la compatibilidad con el HTPP 1.0.
2. **Scripts CGI y FastCGI**. CGI viene de common gateway interface. Los scripts CGI son programas externos que se llaman desde el propio servidor cuando una página lo requiere. El CGI recibe información del servidor web y genera como salida una página web dinámica para el cliente. El script puede realizarse en cualquier lenguaje de programación siempre o que siga las reglas del interfaz CGI. El problema es que es un proceso lento, al tenerse que lanzar un proceso externo al servidor web por cada petición. Perl es uno de los lenguajes más utilizados para ello.
3. **Host virtuales**. Permite atender varios sitios Web en dominios distintos, desde la misma máquina.
4. **Autenticación HTTP**. Permite restringir recursos a determinados usuarios o grupos (distintos de los del sistema).
5. **Intérpretes incluidos en Apache**. Tienen la ventaja sobre los cgi de que están incluidos en el propio Apache y no hay que lanzar un nuevo proceso por cada petición. Los módulos más utilizados son PHP y mod_perl.
6. **Servlets y JSP en Java**. Es una opción que se utiliza en los servidores de aplicaciones, o por ejemplo Tomcat, JBoss, Oracle IAS, WebSphere de IBM o BEA Weblogic. Su gran ventaja será la portabilidad y escalabilidad. Desarrollamos en Java y podemos ejecutarlo en cualquier máquina virtual compatible.
7. **Soporte de SSI** (Server Side Includes) **y de SSL** (Secure Sockets Layer)
8. ...

1.2.1. Arquitectura del servidor Apache.

El servidor Apache es un software que esta estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- Módulos Base: Módulo con las funciones básicas del Apache
- Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, acepando las peticiones y enviando a los hijos a atender a las peticiones
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

1.2.2. Instalación en Mandrake.

Según las opciones seleccionadas en la instalación puede ser que ya se tenga Apache instalado y funcionando, para comprobarlo sólo es necesario abrir un navegador y teclear la dirección http://localhost y si se obtiene la siguiente imagen todo está ya funcionando.

En caso de no obtener la página principal tenemos que instalar Apache, podemos usar rpm desde la línea de comandos o hacerlo desde el centro de control Mandrake.

1.2.2.1. Instalando con rpm.

El módulo que tenemos que instalar es `apache2-2.0.44-11mdk.i586.rpm`[1], pero este módulo tiene dependencias con otros módulos y librerías que tendremos que instalar antes, estos son[2]:

```
libapr0-2.0.44-11mdk.i586.rpm
libtool-1.4.3-1mdk.i586.rpm
libdb4.0-4.0.14-6mdk.i586.rpm
apache2-modules-2.0.44-11mdk.i586.rpm
apache2-common-2.0.44-11mdk.i586.rpm
lynx-2.8.5-0.13mdk.dev.12.i586.rpm
apache-conf-2.0.44-11mdk.i586.rpm
```

Todos esos módulos se encuentran en el primer cd de la distribución, por lo tanto se introduce el cd y se ejecutan las siguientes instrucciones[3]:

```
rpm -i /mnt/cdrom[4]/Mandrake/RPMS/libapr0-2.0.44-11mdk.i586.rpm
```

[1] Se indican las versiones de Apache que se incluyen en la distribución Mandrake 9.1, en otras versiones de Mandrake probablemente sean otras las versiones de Apache.

[2] Las librerías y paquetes que se indican son las necesarias cuando la instalación se realizó dejando las opciones por defecto, en caso de haber realizado la instalación con otras opciones puede ser que alguno de los componentes ya este instalado o que por el contrario se tengan que instalar otros.

[3] Siempre como superusuario.

[4] `La ruta de la unidad de CD-Rom puede cambiar en cada sistema, especialmente cuando se dispone de más de una.`

```
rpm -i /mnt/cdrom/Mandrake/RPMS/libtool-1.4.3-1mdk.i586.rpm
rpm -i /mnt/cdrom/Mandrake/RPMS/libdb4.0-4.0.14-6mdk.i586.rpm
rpm -i /mnt/cdrom/Mandrake/RPMS/apache2-modules-2.0.44-11mdk.i586.rpm
rpm -i /mnt/cdrom/Mandrake/RPMS/apache2-common-2.0.44-11mdk.i586.rpm
rpm -i /mnt/cdrom/Mandrake/RPMS/lynx-2.8.5-0.13mdk.dev.12.i586.rpm
rpm -i /mnt/cdrom/Mandrake/RPMS/apache-conf-2.0.44-11mdk.i586.rpm
rpm -i /mnt/cdrom/Mandrake/RPMS/apache2-2.0.44-11mdk.i586.rpm
```

Después de esto tenemos instalado el servidor Apache en nuestro sistema, pero aun no esta funcionando, para arrancarlo se ejecuta:

```
httpd2 -k start
```

Sino se ha producido ningún error el servidor esta ya funcionando, es posible que al menos se produzca un aviso ya que el servidor no tiene nombre pero eso no impide su arranque. Para comprobarlo abrir el navegador y proceder como se indicó en el apartado anterior.

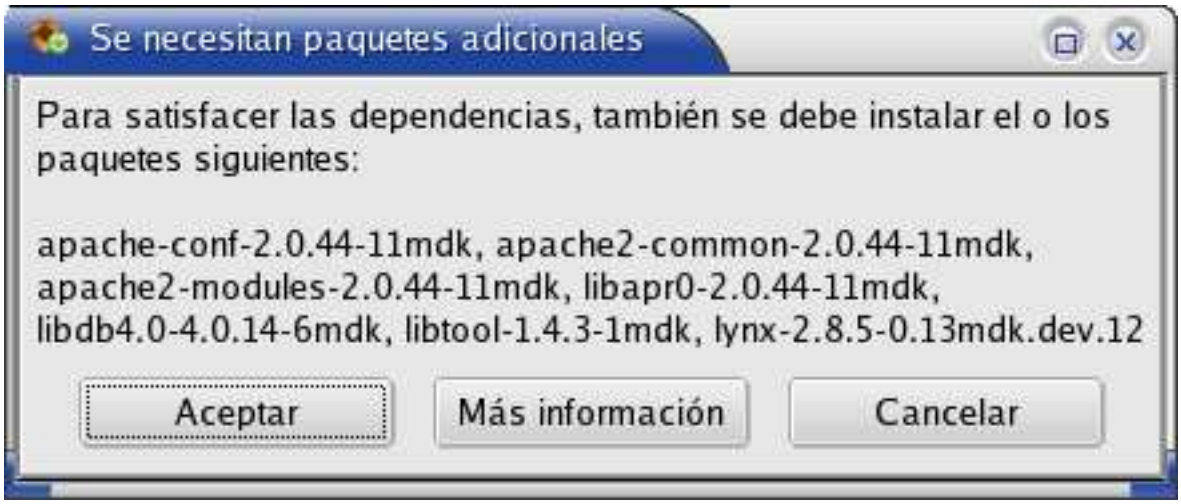
1.2.2.2. Instalación desde el Centro de Control Mandrake.

Si se prefiere se puede realizar la instalación desde el Centro de Control Mandrake, para abrirlo desde el botón de menú KDE se selecciona la opción configuración y lo ejecutamos.

Ahora, se escoge Administración de software y se selecciona "RpmDrake le ayuda a instalar paquetes de software".

Se busca apache y se selecciona el paquete `apache2-2.0.48-5`.

Se pulsa en el botón instalar y se abre una ventana que indica las dependencias y se da opción a instalar los paquetes necesarios.



Se hace clic en aceptar y el servidor se instala, se arranca y se prueba que esta funcionando tal y como se ha indicado en los apartados anteriores.

1.2.3. Instalación en Guadalinex.

1.2.3.1. Instalación utilizando apt-get.

Para instalar Apache en Guadalinex es suficiente con teclear desde una consola, estando como superusuario, la instrucción:

```
apt-get install apache2-mpm-prefork apache-doc apache2-dev
```

Con esto se calcularán las dependencias y se instalarán todos los paquetes necesarios, además se iniciará el demonio y se incluirá en el script de arranque. Si este proceso tuviera algún problema, la instrucción para iniciar el demonio manualmente es:

/etc/init.d/apache2 start

Y para incluir el demonio en el script de arranque:

update-rc.d apache2 defaults

1.2.3.2. Instalación con el gestor de paquetes Debian.

Para acceder al gestor de paquetes, desde el menú *Aplicaciones* se selecciona *configuración*, después *sistema* y por último *Synaptic (Gestor de paquetes)*

Para seleccionar un paquete se hace clic con el boton derecho sobre él y se escoge: *Marcar para instalación*.

Los paquetes que se deben seleccionar son[5]:

- apache2
- apache2-common
- apache2-mpm-worker
- libapr0
- ssl-cert

Una vez seleccionados todos los paquetes basta con hacer clic en *Aplicar* ... y esperar.

[5] Es posible que alguno se marque por dependencias.

1.2.3.3. Instalación desde un fichero tar.gz.

Las instalaciones anteriores necesitan tener conexión a Internet para realizar la descarga de los paquetes, si alguien no dispone de conexión puede optar por, desde otra máquina, descargar el paquete[6] y realizar la compilación del mismo para instalarlo.

La página oficial de Apache para realizar la descarga es *www.apache.org*, otra página para realizar la descarga de forma más sencilla es *www.softonic.es*.

Una vez obtenido el fichero:

1. Descomprimir el paquete.
2. En el directorio donde se ha descomprimido el fichero:
   a) `./configure --prefix=PREFIX`[7]
   b) `make`
   c) `make install`
   d) `PREFIX/bin/apachectl start`

Y ya lo tenemos funcionando.

1.2.3.4. Instalación con el CD de Suplementos para Guadalinex 2004.

Esta opción es la menos recomendable pues se instala la versión 1.3 del servidor.

Desde el menú *Aplicaciones* se selecciona *Configuración* y *GMAX (Instalación de suplementos)*.

En la lista *Seleccione un grupo* se escoge *servidores* y en *programa* se coge *apache,* se hace clic en instalar.

1.2.4. Los ficheros de configuración.

En Mandrake el fichero de configuración principal de Apache se encuentra en la siguiente ubicación `/etc/httpd/conf/httpd2.conf`, hay que notar que este fichero incluye unas directivas Include que lo que hacen es incluir otros ficheros de configuración, en concreto se incluye el archivo `/etc/httpd/conf/commonhttpd.conf`, todos los ficheros del directorio `/etc/httpd/conf.d` y el fichero `/etc/httpd/conf/vhost/Vhost.conf`.

Una configuración un tanto extraña, pero que tiene su explicación:

En el fichero commonhttpd.conf se ponen las directivas comunes con otras versiones de Apache.

En el directorio conf.d se colocan los archivos de configuración de módulos particulares de la versión 2.0.

El fichero Vhost.conf incluye la configuración de los host virtuales.

En Guadalinex los ficheros de configuración los tenemos en `/etc/apache2` siendo `apache2.conf` el fichero principal. Si se ha realizado la instalación desde un tar.gz dependerá del lugar donde se ha realizado[8].

A partir de aquí trabajaremos con los ficheros de configuración de Mandrake, para hacerlo con Guadalinex utilizar el fichero antes indicado.

Tanto el archivo httpd2.conf como el commonhttpd.conf están bien comentados y son bastante autoexplicativos. La configuración predeterminada funciona para los ordenadores de la mayoría de los usuarios, así que probablemente no necesitará cambiar ninguna de las directivas en los ficheros. Sin embargo, es bueno a conocer las opciones de configuración más importantes.

Antes de modificar un fichero de configuración es conveniente copiar el fichero original, por ejemplo, el fichero httpd2.conf dándole el nombre httpd2.conf.copia u otro que nos sea significativo. De esta manera si cometemos un error mientras estamos modificando el fichero de configuración, no debemos preocuparnos porque siempre dispondremos de una copia de seguridad.

Si cometemos un error y nuestro servidor web no funciona correctamente, el primer sitio donde acudir es a lo que acabamos de modificar en httpd2.conf o commonhttpd.conf. Después podemos consultar el

[6] En Mandrake estamos trabajando con la versión 2.0.48, la última disponible es la 2.0.52.

[7] Sustituir PREFIX por la ruta donde se desea realizar la instalación, usualmente se utiliza /usr/local/apache2

[8] Si se utilizó el directorio /usr/local/apache2 el fichero de configuración estará en /usr/local/apache2/conf/.

fichero de error (/var/log/httpd/error_log), las ultimas entradas deberán servirnos de ayuda para saber lo que ha pasado.

Las líneas que comienzan con # se consideran comentarios y el servidor no las tiene en cuenta.

A continuación se dan breves descripciones de las directivas:

ServerRoot El comando ServerRoot se va a referir al directorio principal donde se encuentran todos los ficheros de configuración y trabajo del servidor. Su valor es /etc/httpd.

ServerAdmin: especifica la dirección de correo electrónico del administrador, esta dirección aparece en los mensajes de error, para permitir al usuario notificar un error al administrador. No puede estar dentro de ninguna sección.

ServerName: especifica el nombre y el puerto que el servidor utiliza para identificarse, normalmente se determina automáticamente, pero es recomendable especificarlo explícitamente para que no haya problemas al iniciar el servidor. Si el servidor no tiene un nombre registrado en las DNS, se recomienda poner su número IP. No puede estar dentro de ninguna sección.

La sintaxis es:

```
ServerName direccionIP:Puerto
```

por ejemplo:

```
ServerName localhost:80
```

User La directiva User establece el userid que utiliza el servidor para ejecutarse y responder a las peticiones. El valor de User determina el acceso que tendrá el servidor a los ficheros a y directorios con las páginas. Cualquier fichero al que no pueda acceder este usuario, será inaccesible para el servidor web y como consecuencia, también inaccesible al visitante de la web. El comando predeterminado para User es apache. El usuario User también es dueño de cualquier proceso CGI que arranque el servidor y no se le deberá permitir ejecutar ningun código que no esté pensado para responder peticiones HTTP. El proceso httpd padre se inicia como root durante operaciones normales, pero pasa al usuario apache inmediatamente. El servidor debe arrancar como root porque necesita un puerto por debajo de 1024 (el puerto por defecto es el 80). Los puertos por deba jo de 1024 están reservados para el sistema, así que sólo se pueden usar si se es root. Una vez que el servidor se ha conectado al puerto, pasa el proceso a User antes de aceptar peticiones.

Group El comando Group es similar a User. Group establece el grupo en el que el servidor responde a las peticiones. El valor predeterminado del comando Group también es apache, en este caso como grupo, y no como usuario.

DocumentRoot DocumentRoot es el directorio que contiene la mayoría de los archivos HTML que se entregarán en respuesta a peticiones. El directorio predeterminado DocumentRoot a es /var/www/html.
Por ejemplo, el servidor puede recibir una petición para el siguiente o documento: http://localhost/prueba.html, el servidor buscará el fichero en el siguiente directorio por defecto: /var/www/html/prueba.html

DirectoryIndex: especifica el fichero por defecto que buscará en cada directorio, en caso de que no se especifique ninguno.

En esta directiva se pueden especificar más de un fichero, la sintaxis es la siguiente:

```
DirectoryIndex fichero1 fichero2 fichero3
```

El orden con el que se especifica el nombre de fichero determinará la prioridad a la hora de decidir que fichero es el que se muestra.

La directiva se puede encontrar fuera de cualquier sección, dentro de una sección o dentro de un fichero .htaccess.

AccessFileName: es el nombre del fichero de configuración que se buscará en cada una de los directorios del servidor para conocer la configuración del mismo. Este fichero permite configurar el comportamiento de cada uno de los directorios individualmente. Para que esta configuración funcione, la

directiva AllowOverride tiene que tener un valor que lo permita. No puede estar dentro de ninguna sección.

El nombre de fichero que se especifica por defecto es el del fichero ".htaccess".

Como medida de seguridad, la configuración del Apache establece que no se muestre la existencia de este fichero a ningún usuario, aunque este establecida la opción de listado de directorios. Si se decide cambiar al nombre, habrá que redefinir la seguridad para que no se muestre el contenido del nuevo fichero. Esto se hace en el fichero commonhttpd.conf en una sección File como la que se presenta a continuación en la que se establece que todos los ficheros que comiencen por .ht no se mostrarán.

```
<Files ~ "^\.ht">
   Order allow,deny
   Deny from all
</Files>
```

PidFile Ubicación del fichero que contendrá el número de identificación del proceso cuando se encienda el servidor.

TimeOut el valor se utiliza para configurar medido en segundos, tres parámetros:

1. El tiempo tal que puede tardar una petición en ser recibida entera
2. La cantidad de tiempo que espera entre recepción de paquetes TCP
3. La cantidad de tiempo entre ACK's en transmisiones TCP

Pasado este tiempo se produce un mensaje de error en el que se indica que se ha consumido el tiempo máximo de espera. Establecer un valor muy pequeño puede dar lugar a que los usuarios reciban este mensaje de error, y establecer un valor muy pequeño dará lugar a una sobrecarga de la máquina.

KeepAlive especifica si se utilizarán conexiones persistentes, es decir, que todas las peticiones de un usuario se atenderán con la misma conexión.

MaxKeepAliveRequests número máximo de conexiones persistentes. (número máximo de usuarios concurrentes si KeepAlive esta en ON). Para establecer este parámetro, hay que tener en cuenta el ancho de banda de salida de nuestro servidor, por el cual deberá ser enviada toda la información, si se establece un valor muy grande respecto al ancho de banda, el tiempo de respuesta se verá incrementado para cada usuario.

KeepAliveTimeout: tiempo que espera en segundos entre peticiones de un usuario, antes de considerar que este ha terminado, y cerrar su conexión.

Si el valor es muy pequeño provocará que algunos usuarios no puedan visualizar la página debido a que el número máximo de conexiones persistentes se ha superado, mientras que si se establece un valor muy grande se estarán utilizando muchos recursos de la máquina.

Listen: esta directiva permite especificar que puerto se utilizará para antender las peticiones. Por defecto se utiliza el puerto 80 (www), también permite especificar que direcciones IP atenderá, por defecto todas. Para atender dos direcciones IP distintas, con distintos puerto, se utilizaría:

Listen 192.168.255.5:80

Listen 192.168.255.8:8080

LoadModule Directiva que sirve para cargar módulos que incluyen distintas funcionalidades. La sintaxis es:

LoadModule nombreModulo ubicacionFichero

Directory Las etiquetas <Directory /path/a/directorio>y </Directory>se usan para agrupar directivas de configuración que se aplican a ese directorio y sus subdirectorios. Cualquier directiva aplicable a un directorio puede usarse en las etiquetas <Directory>. Las etiquetas <File>pueden aplicarse de la misma forma a un fichero específico.

Por defecto, se aplican parámetros muy restrictivos al directorio raíz (/).

```
<Directory />
   Options FollowSymLinks
   AllowOverride None
</Directory>
```

Con la directiva Options establecemos qué características están disponibles para el directorio en el que se establece, su sintaxis es: Options [+|-]opcion [+|-]opcion ...

y las opciones pueden ser

All Todas las opciones excepto para MultiViews. Es el entorno por defecto.

ExecCGI Se permite ejecutar scripts CGI usando mod_cgi.

FollowSymlinks El servidor seguirá los enlaces simbólicos en este directorio. Aunque el servidor siga los enlaces simbólicos, no cambia el nombre de path usado para comparar las secciones <Directory>. Esta opción se ignora si se establece dentro o de una sección <Location>.

Includes Se permiten inclusiones del lado del servidor proporcionadas por mod_include.

IncludesNOEXEC Se permiten inclusiones del lado del servidor, pero están desactivados #exec cmd y #exec cgi. Está activo para scripts CGI #include virtual desde directorios ScriptAlias.

Indexes Si hay una petición de una URL de un directorio y en él no hay DirectoryIndex (ej: index.html), mod_autoindex devolverá un listado formateado del directorio.

MultiViews Los contenidos negociados "MultiViews" se permiten usando mod_negotiation.

SymLinksIfOwnerMatch El servidor sólo seguirá los enlaces simbólicos para aquellos archivos o directorios que posean la misma identidad de usuario que el enlace. Esta opción se ignora si se establece dentro de una sección <Location>.

Normalmente, si se pueden aplicar varias Options a un directorio sólo se usa la más específica, ignorándose las demás; las opciones no se mezclan. En cualquier caso, si todas las opciones de la directiva Options van precedidas por el símbolo + o -, se mezclarán. Cualquier opción precedida por + se anadirá a las opciones en vigor, y cualquiera precedida por -, se eliminará. ~ a a Tal cual está, es equivalente a:

```
Options FollowSymLinks -ExecCGI -Includes -Indexes -Multiviews
```

e implicará que está permitido atravesar los enlaces simbólicos en todo el sistema.

Con AllowOverride a None establecemos que el servidor no leerá el archivo especificado en AccessFileName (.htaccess). Esta directiva permite especificar qué partes del servidor pueden ser establecidas en los archivos .htaccess, los valores que puede tomar (además del comentado) son:

AuthConfig permite el uso de directivas de autorización (por ejemplo: AuthName, AuthType, o Require, ...)

FileInfo permite el uso de directivas que establecen el tipo de documento (por ejemplo: DefaultType, ErrorDocument, ...)

Indexes permite usar directivas para controlar la forma en que se realizan los índices de los directorios (por ejmplo: AddIcon, IndexOptions, etc)

Limit permite el uso de directivas para establecer el control de acceso (Allow, Deny y Order)

Options permite usar directivas que controlan opciones espec´ficas del directorio (Options i y XBitHack)

Con esta configuración, cualquier directorio del sistema que necesite valores más permisivos ha de ser configurado explícitamente. El directorio cgi-bin está configurado para permitir la ejecución de scripts CGI, con la opción ExecCGI. Si se necesita ejecutar un script CGI en cualquier otro directorio, habrá que configurar ExecCGI para ese directorio. Por ejemplo, si cgi-bin es /var/www/cgi-bin, pero se quieren ejecutar scripts CGI desde /home/usuario/cgi-bin, anadirá una directiva ExecCGI a un par de directivas Directory como las siguientes:

```
<Directory /home/usuario/cgi-bin>
   Options +ExecCGI
</Directory>
```

Para permitir la ejecución de scripts CGI en /home/usuario/cgi-bin, habrá que llevar a cabo pasos extra aparte de configurar ExecCGI. El valor de los permisos para scripts CGI y el recorrido entero a los scripts, debe ser de 0755. Además, el dueño del script y del directorio deben ser el mismo.

UserDir UserDir es el nombre del subdirectorio, dentro del directorio de cada usuario, donde estarán los archivos HTML que serán servidos. Por defecto, el subdirectorio es public_html. Por ejemplo, el servidor podría recibir la siguiente petición:

```
http://localhost/~usuario/prueba.html
```

El servidor buscaría el fichero:

```
/home/usuario/public_html/prueba.html
```

En el ejemplo, /home/usuario es el directorio del usuario.

Hay que asegurarse que:

- Los permisos sean los adecuados:
  - De los directorios de usuario sean correctos (711).
  - Los bits de lectura (r) y ejecución (x) deben estar activados en el directorio public_html (755 valdrá).
  - El valor de los permisos con que se servirán los ficheros desde public_html debe ser 644 por lo menos.
- Permitir el acceso usando el módulo mod_userdir. Con él conseguimos que Apache permita o deniegue esta forma de acceso. Así, si deseamos activar esta posibilidad, hemos de cambiar la sección como sigue:

  ```
  <IfModule mod_userdir.c>
     # UserDir disable
     UserDir public_html
  </IfModule>
  ```

  [9]

  trás modificar un fichero de configuración es importante tener en cuenta que se debe recargar la configuración.[10]

ErrorLog ErrorLog nombra el fichero donde se guardan los errores del servidor. Por defecto, el fichero de error del servidor es /var/log/httpd/error_log. El log de errores es un buen sitio para detectar problemas en el servidor. Una línea de ejemplo puede ser: `[Sun Mar 21 05:39:18 2004] [error] [client 66.90.73.73] File does not exist: /var/www/html/sumthin`

CustomLog con esta directiva establecemos la ubicación y formato del archivo de registro de acceso. Por defecto:

```
CustomLog log/access.log "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{UserAgent}i\""
```

es decir:

[9] Si modifica el nombre del directorio de usuario tenga en cuenta que en los ficheros de configuración existen otras clausulas <Directory ...>que modifican la configuración del directorio, por lo que será necesario modificarlas también.

[10] Para recargar la configuración en Mandrake `httpd -k restart`, en Guadalinex `/etc/init.d/apache2 restart`.

- Registramos el host remoto (%h), la identidad del cliente (%l), si se necesita autenticación para la URL solicitada, el nombre de usuario (%u) y el tiempo de solicitud (%t).
- Se entrecomilla la primera línea de la solicitud (%r), almacenamos el estado devuelto por el servidor en respuesta a la solicitud (%>s) y los bytes enviados (%b).
- Además de la cabecera enviada por el cliente al solicitar la página web, almacenamos la URL de la página solicitada (%{Referer}i) y el navegador web usado (%{User-Agent}i).

Una línea de ejemplo:

```
127.0.0.1 - juan [25/Oct/2004:20:58:05 +0200] "GET /~pedro HTTP/1.1" 301 361 "-" "Mozilla/5.0
(X11; U; Linux i686; es-ES; rv:1.6) Gecko/20040115"
```

Para practicar montar el servidor web Apache y comprobar que los usuarios del sistema pueden acceder a sus páginas web personales. Supongamos que en nuestra máquina hay un usuario de nombre cervantes.

1. Para el usuario cervantes crear el directorio $HOME/public_html

   ```
   $mkdir public_html
   ```

2. Poner en él un fichero html simple de nombre index.html, por ejemplo:

   ```
   <html>
      <body>
         <h1>Esta es la web de Cervantes</h1>
      </body>
   </html>
   ```

3. Modificar los permisos del $HOME de cervantes, así como del directorio public_html para que Apache pueda acceder a él:

   ```
   $chmod 711 $HOME
   $chmod 755 $HOME/public_html
   ```

4. Permitir que Apache acceda a directorios de usuario mediante http://servidor_web/~usuario. Para ello cambiemos, si es necesario, la sección del fichero de configuración del servidor como sigue:

   ```
   <IfModule mod_userdir.c>
      # UserDir disable
      UserDir public_html
   </IfModule>
   ```

   y recargar después la configuración del servicio con:

   ```
   #httpd2 -k restart
   ```

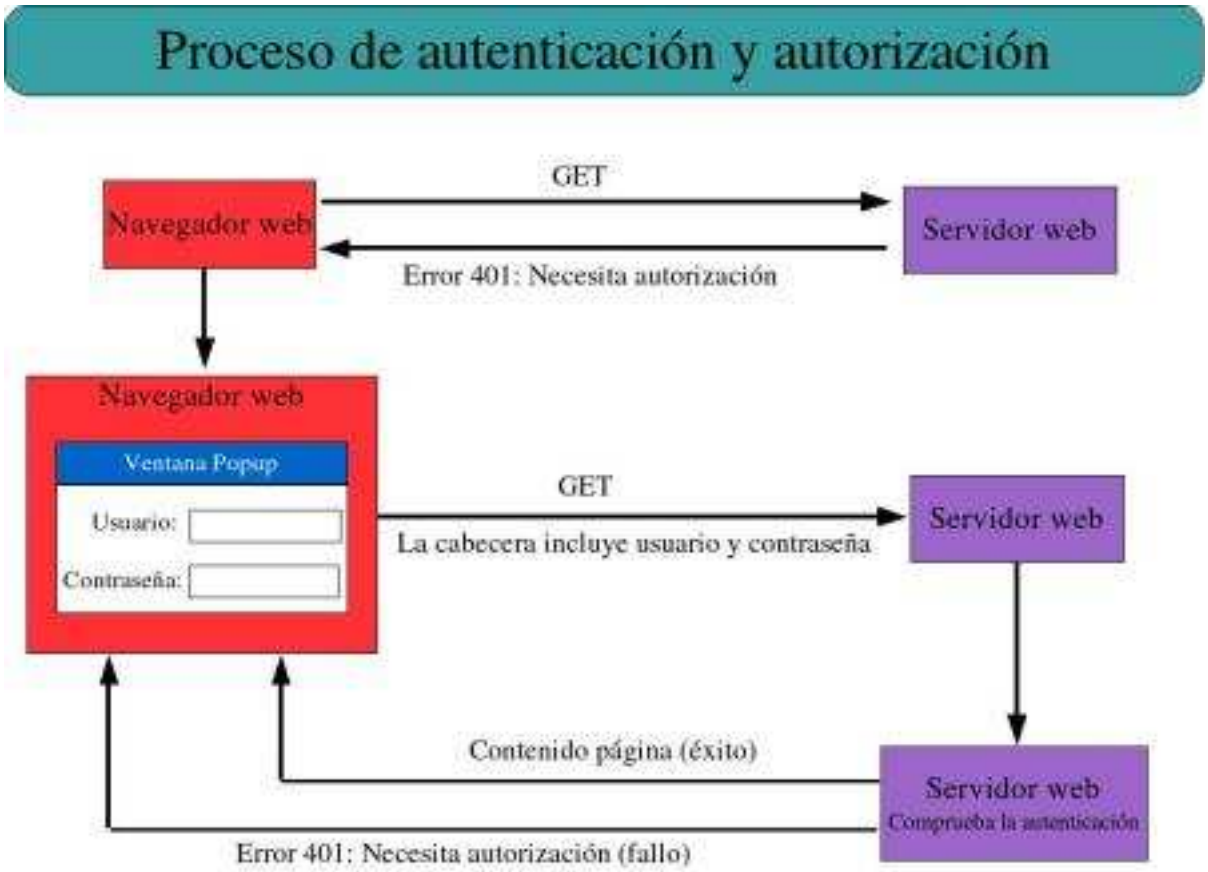
5. Comprobar que funciona cargando con nuestro navegador la página web http://localhost/~cervantes

1.2.5. Autenticación.

Podemos conseguir que para acceder a determinados recursos un cliente tenga que autenticarse ante el servidor. La información contenida en esa zona sólo deberá ser vista por el usuario o grupo de usuarios que establezcamos.

Apache tiene varios módulos que permiten realizar la autenticación de usuarios, el módulo por defecto es mod_auth. Mod_auth almacena las credenciales en ficheros de texto plano, si bien las contraseñas se guardan encriptadas. El uso de este módulo está recomendado para sistemas que no superan los cien usuarios.

El proceso de autenticación es simple. El navegador solicita la página, el servidor mediante un código 401 solicita el identificador y la contraseña y el cliente envía su nombre y contraseñas. A continuación Apache comprueba su archivo de nombres y contraseñas cifradas para ver si el cliente tiene derecho a acceder, en caso afirmativo envía el documento solicitado y en caso contrario vuelve a mandar un código 401 solicitando de nuevo la autenticación y la contraseña.



Podemos establecer de dos formas diferentes la autenticación:

- Globalmente: agregando una sección <Directory /path/a/directorio>y </Directory>en nuestro archivo de configuración por cada directorio que deseemos proteger. Por ejemplo con:

```
<Directory /var/www/html/privado>
   AuthType Basic
   AuthName "Pagina de Cervantes"
   AuthUserFile /var/www/passwd/.htpasswd
   require user cervantes
</Directory>
```

- Usando los ficheros especiales .htaccess[11]. Las directivas que se pongan dentro de los ficheros .htaccess[12] se aplicarán sólo al directorio que lo contiene, así como a todos sus subdirectorios. Los archivos .htaccess se leen cada vez que hay una petición de páginas y, por tanto, no hay que reinicair el servidor Web para que se activen los cambios que realicemos en ellos.

Las directivas de autenticación del módulo mod_auth (módulo de autenticación por defecto de Apache) son:

AuthUserFile asigna el nombre del archivo de texto que contendrá los nombres de usuario y a contraseñas usadas en la autenticación HTTP básica.

[11] Se puede cambiar el nombre de estos ficheros modificandolo en la directiva FileAccessName
[12] Las mismas que se han puesto dentro de la clausula <Directory ...>en el fichero de configuración.

AuthGroupFile asigna el nombre del archivo de texto que contendrá la lista de grupos de usuarios usadas en la autenticación HTTP básica. Una línea de este archivo (en él se crea un grupo de nombre curso con tres usuarios) puede ser

```
curso: alum1 alum2 alum3
```

AuthAuthoritative toma los valores On y Off (por defecto está en On). Permite que si usamos en un directorio varios métodos de autenticación diferentes y falla el primero, se pase al segundo.

Las demás directivas utilizadas son:

AuthType siempre se fija en Basic porque es la que HTTP soporta por defecto.

AuthName crea la etiqueta en la ventana que el navegador web mostrará a los usuarios para solicitar su identificación.

require exige que sólo tengan acceso los usuario o grupos que se indican.

Vamos a crear un directorio con acceso restringido al usuario cervantes

1. Creemos el directorio:

   ```
   # mkdir /var/www/html/privado
   ```

   y pongamos en él una página web simple de nombre index.html.

2. Creemos el directorio[13] en donde almacenar las claves de acceso, por ejemplo:

   ```
   # mkdir /var/www/passwd
   ```

   Hay varias formas de trabajar con archivos de contraseñas. Como ya se ha indicado con menos de 100 usuarios se puede trabajar con archivos planos, en otros casos habría que utilizar el módulo mod_auth_mysql. Para archivos planos se usa la herramienta htpasswd, en Guadalinex htpasswd2.

   ```
   # htpasswd -c /var/www/passwd/.htpasswd cervantes
   ```

   De esta forma se crea (-c) el archivo con el primer usuario y se nos pedirá la contraseña (hacer notar que no tiene por qué ser un usuario del sistema). Después, para añadir otros usuarios el parámetro -c no hay que ponerlo.

3. Creemos en `/var/www/html/privado` el fichero `.htaccess` con el siguiente contenido:

   ```
   AuthType Basic
   AuthName "Pagina restringida de Cervantes"
   AuthUserFile /var/www/passwd/.htpasswd
   require user thales
   ```

   Comentemos un poco el fichero: con la directiva AuthType con el valor Basic indicamos que la contrasena se negociará en texto plano. En el cuadro de verificación de contraseña, veremos el texto "Página restringida de Cervantes". Por ultimo indicamos a Apache el archivo en donde buscar la contraseña y que el nombre de usuario requerido es cervantes.

4. Modifiquemos el fichero /etc/httpd/conf/commonhttpd.conf[14]. Si desde nuestro navegador web intentamos cargar la página:

   ```
   http://127.0.0.1/public/index.html
   ```

   podremos cargarla sin problema. Esto se debe a que en el fichero /etc/httpp/conf/http.conf hay una sección como la que sigue:

   ```
   <Directory />
      Options -All -Multiviews
   ```

[13] Para garantizar la seguridad es importante que el directorio se cree fuera del DocumentRoot de Apache.
[14] En Guadalinex la configuración está realizada mediante un host virtual en el directorio *sites-available* en el fichero *default.*

```
    AllowOverride None
    <IfModule mod_access.c>
       Order deny,allow
       Deny from all
    </IfModule>
</Directory>
```

La línea :

```
AllowOverride None
```

hace que los ficheros .htacces no pueden modificar nada. Al poner esta otra:

```
AllowOverride AuthConfig
```

permitimos que controlen la autenticación.

Las directivas dentro de la sección:

```
<IfModule mod_access.c>
```

Controlan quién puede obtener la respuesta de este servidor. Tal cual está primero se procesan primero las directivas deny y depués las allow, además, la seguna línea:

```
Deny from all
```

deniega el acceso a todo el mundo, por lo que se debe sustituir por:

```
Allow from all
```

5. Por último y como se han modificado ficheros de configuración tan sólo queda reiniciar el servidor.

1.2.5.1. Autenticación usando directivas de grupo.

En el apartado anterior, cuando mediante la clausula require se ha determinado el usuario al que se le permite el acceso, se ha utilizado:

```
requiere user cervantes
```

si se le quiere dar acceso a otro usuario, además de crearlo con la utilidad htpasswd, se tendrá que añadir a la clausula require:

```
requiere user cervantes calderon
```

Para evitar esta modificación se utiliza:

```
requiere valid-user
```

con la que se consigue que puedan acceder al directorio todos los usuarios definidos en el fichero de usuario - contraseña.

Pero, en el caso de tener varios directorios de acceso restringido vuelve a surgir el problema, veamoslo con un ejemplo:

En un centro se crearán dos secciones dentro del sitio web, una de acceso exclusivo a profesores y otra de acceso exlusivo a alumnos.

En esta situación no es imposible utilizar require valid-user, a menos que se creen dos ficheros de usuario - contraseña diferenciados con lo que las secciones de configuración quedarían:

```
<Directory /var/www/html/profesores>
   AuthType Basic
   AuthName "Pagina de Profesores"
   AuthUserFile /var/www/passwd/.profes
   require valid-user
</Directory>

<Directory /var/www/html/alumnos>
   AuthType Basic
   AuthName "Pagina de Alumnos"
   AuthUserFile /var/www/passwd/.alumnos
   require valid-user
```

```
</Directory>
```

Normalmete, la situación real de cualquier sitio web es que los usuarios no tengan perfiles tan definidos y puedan acceder a varias secciones. En nuestro ejemplo se podría plantear que los profesores pudieran acceder a la sección de alumnos, de esta forma estaríamos obligados a crear a los profesores como usuarios en el fichero de alumnos, duplicando de esa forma el trabajo. Aquí entran las directivas de grupo.

Se crea un único archivo de usuario - contraseña y se modifica la congiguración:

```
<Directory /var/www/html/profesores>
   AuthType Basic
   AuthName "Pagina de Profesores"
   AuthUserFile /var/www/passwd/.miembros
   AuthGroupFile /var/www/passwd/.grupos
   require group profes
</Directory>

<Directory /var/www/html/alumnos>
   AuthType Basic
   AuthName "Pagina de Alumnos"
   AuthUserFile /var/www/passwd/.miembros
   AuthGroupFile /var/www/passwd/.grupos
   require group alumnos
</Directory>
```

Como se observa se han añadido las cláusulas AuthGroupFile que indican donde se encuentra el fichero de definición de grupos, que por cierto aun no hemos creado. El fichero de grupos es un archivo de texto con la siguiente estructura:

```
nombreGrupo1: usuario1 usuario2 ....
nombreGrupo2: usuario3 usuario4 ....
```

En nuestro ejemplo podría ser:

```
profes: prof1 prof2 prof3
alumnos: alum1 alum2 alum3 prof1 prof2 prof3
```

Por último, en la clausula require se especifica el grupo que tiene acceso a la sección.

De esta forma se pueden añadir usuarios con el comando htpasswd, modificar el fichero de grupos y no se necesita reiniciar el servidor ni modificar los ficheros de configuración y .htaccess.

1.2.6. Host Virtuales.

1.2.6.1. Características.

1. Permite alojar varios dominios en una sola máquina.
2. Permite crear subdominios.
3. Se pueden basar en IP o en nombre.
4. Un host virtual hereda toda la configuración del sitio principal, a no ser que se redefina.

1.2.6.2. Host virtuales basados en nombre.

En *httpd2.conf* encontramos una directiva:

```
include conf/vhosts/Vhosts.conf
```

Al editar el fichero encontramos dos configuraciones de ejemplo, una basada en IP y otra en nombre, pero las dos aparecen comentadas, por lo que podemos tomarlas como referencia para hacer las nuestras en este mismo fichero para tenerlas identificadas.

Veamos ahora un ejemplo de una configuración de dos host virtuales basadas en nombre, haremos que nuestra máquina responda a peticiones de dos dominios distintos sobre la misma IP:

```
NameVirtualHost 192.168.2.2
<VirtualHost 192.168.2.2>
    ServerName servidor1.miservidor.org
```

```
    DocumentRoot /var/www/html/servidor1
</VirtualHost>
<VirtualHost 192.168.2.2>
    ServerName servidor2.miservidor.org
    DocumentRoot /var/www/html/servidor2
</VirtualHost>
```

La directiva NameVirtualHost especifica la dirección IP sobre la que se van a montar los host virtuales, se utilizan también contenedores <VirtualHost></VirtualHost>, todo lo que se incluya en ellos sólo afectará al host virtual.

ServerName especifica el dominio y *DocumentRoot* el directorio raíz del host virtual, evidentemente se puede utilizar dentro del host virtual cualquier otra directiva de Apache.

Usando el fichero /etc/hosts

Bueno, ahora no puedo cargar los hosts virtuales utilizando localhost, necesitaría un DNS que apuntará mis dominios *servidor1.miservidor.org* y *servidor2.miservidor.org* a mi dirección IP 192.168.2.2.

Como de momento no sabemos configurar un DNS vamos a solucionarlo de otra forma, para eso editamos el fichero */etc/nsswitch.conf* en el encontramos una línea:

```
hosts: files nisplus nis dns
```

Esta línea le dice a linux como tiene que resolver los nombres de dominio de los hosts, en primer lugar mira si está en los ficheros, sino está lo busca en nisplus y nis, y finalmente en el DNS.

Es decir, el primer sitio que se va a mirar para resolver el nombre de una máquina es en el fichero /etc/hosts, si lo editamos encontramos dos líneas:

127.0.0.1 localhost
127.0.0.1 nuevo

Normalmente contendrá la dirección de loopback asociado a localhost y al nombre que se le dió a la máquina en la instalación.

Ahora añadiremos dos líneas para que resuelva los hosts virtuales que hemos creado[15]:

```
192.168.2.2 servidor1.miservidor.org
192.168.2.2 servidor2.miservidor.org
```

Con esto conseguimos que la máquina local identifique como propios los dominios, para que otra máquina pueda acceder a ellos tendrá que poner las mismas líneas en su fichero.

1.2.6.3. Host virtuales basados en IP.

Ahora se trata de montar los host virtuales sobre direcciones IP distintas, el ejemplo anterior se realizaría de la forma siguiente:

```
<VirtualHost 192.168.2.10>
    ServerName servidor1.miservidor.org
    DocumentRoot /var/www/html/servidor1
</VirtualHost>
<VirtualHost 192.168.2.11>
    ServerName servidor2.miservidor.org
    DocumentRoot /var/www/html/servidor2
</VirtualHost>
```

Como se puede observar en los contenedores se colocan las IP's asociadas a cada dominio y no se utiliza en este caso la directiva *NameVirtualHost*.

Evidentemente se tienen que realizar los cambios en el fichero */etc/hosts*.

[15] Antes de cambiar un fichero de configuración se debe realizar una copia de seguridad del mismo.

Alias de interfaces de red.

Si nuestra máquina no tiene nada más que una interfaz de red no podemos utilizar esta configuración, a menos que utilicemos alias.

Para ver los interfaces de red de una máquina se utiliza el comando ifconfig, ya comentado, que da como resultado:

```
[root@nuevo vhosts]# ifconfig
eth0 Link encap:Ethernet HWaddr 00:05:1C:11:B4:4A
    inet addr:192.168.2.2 Bcast:192.168.2.255 Mask:255.255.255.0
    inet6 addr: fe80::205:1cff:fe11:b44a/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:2274 errors:0 dropped:0 overruns:0 frame:0
    TX packets:3241 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:1397634 (1.3 Mb) TX bytes:393427 (384.2 Kb)
    Interrupt:11 Base address:0xe000
lo   Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING MTU:16436 Metric:1
    RX packets:933 errors:0 dropped:0 overruns:0 frame:0
    TX packets:933 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:221170 (215.9 Kb) TX bytes:221170 (215.9 Kb)
```

La tarjeta eth0 corresponde al intefaz físico y lo corresponde al loopback.

Para añadir alias de eth0 se utiliza el mismo comando:

```
ifconfig eth0:0 192.168.2.10 up
ifconfig eth0:1 192.168.2.11 up
```

Se ha creado un alias de la eth0 llamado eth0:0 asociado a la IP 192.168.2.10 y se activado con la opción up, e igual para la eth0:1 en la IP 192.168.2.11.

Si ahora se vuelve a ejecutar ifconfig sin parámetros para que muestre la información de todos los interfaces se obtiene:

```
[root@nuevo vhosts]# ifconfig
eth0 Link encap:Ethernet HWaddr 00:05:1C:11:B4:4A
    inet addr:192.168.2.2 Bcast:192.168.2.255 Mask:255.255.255.0
    inet6 addr: fe80::205:1cff:fe11:b44a/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:2348 errors:0 dropped:0 overruns:0 frame:0
    TX packets:3333 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:1416007 (1.3 Mb) TX bytes:401401 (391.9 Kb)
    Interrupt:11 Base address:0xe000
eth0:0 Link encap:Ethernet HWaddr 00:05:1C:11:B4:4A
    inet addr:192.168.2.10 Bcast:192.168.2.255 Mask:255.255.255.0
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    Interrupt:11 Base address:0xe000
eth0:1 Link encap:Ethernet HWaddr 00:05:1C:11:B4:4A
    inet addr:192.168.2.11 Bcast:192.168.2.255 Mask:255.255.255.0
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    Interrupt:11 Base address:0xe000
lo Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING MTU:16436 Metric:1
    RX packets:935 errors:0 dropped:0 overruns:0 frame:0
    TX packets:935 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:221438 (216.2 Kb) TX bytes:221438 (216.2 Kb)
```

Para anular los alias se utiliza el mismo comando con la opición down:

```
ifconfig eth0:0 down
ifconfig eth0:1 down
```

1.2.7. Servidores seguros.

1.2.7.1. Características.

1. Funcionan con el protocolo SSL.
2. Las direcciones se escriben precedidas de https.
3. La diferencia es que la información viaja por la red cifrada.
4. Se tiene que instalar el módulo *mod_ssl* y el páquete *openssl*.

1.2.7.2. Configuración.

Por defecto está configurado sobre el *DocumentRoot* del servidor, por lo tanto se puede probar con la dirección: *https://localhost*

En *httpd2.conf* encontramos una directiva:

```
Include /etc/httpd/conf.d/*.conf
```

Está línea está incluyendo todos los ficheros de configuración del directorio */etc/httpd/conf.d*

En ese directorio existen dos ficheros que afectan a la configuración del servidor seguro *40_mod_ssl.conf* que realiza la configuración general del módulo y el fichero *41_mod_ssl.default-vhost.conf*.

El segundo define un host virtual sobre el que se establece el servidor seguro, no se necesita cambiar gran cosa, simplemente el *DocumentRoot* y el *ServerName*.

El 443 es el puerto por el que se escuchan las peticiones.

En Guadalinex resulta bastante más complicado, por lo que se va a detallar el proceso.

La configuración la vamos a realizar sobre a o el servidor web www.midominio.com.

En primer lugar activemos el módulo mod_ssl con[16]:

```
# a2enmod ssl
```

y modifiquemos el fichero */etc/apache2/ports.conf*

```
Listen 80
Listen 443
```

para que Apache escuche en el puerto 443.

Para no tener que cambiar demasiado los ficheros de configuración crearemos el directorio */etc/apache/sites* y nos situamos en él: mkdir /etc/apache2/sites

cd /etc/apache2/sites

Ejecutemos los comandos:

```
for i in 1 2 3 4 5 6 7 8 9; do dd if=/dev/urandom count=2 bs=1k | od a| sed
        -e 's/......//' >rand$i; gzip -1 rand$i; done
openssl genrsa -des3 -rand
        rand1.gz:rand2.gz:rand3.gz:rand4.gz:rand5.gz:rand6.gz:rand7.gz:rand8.gz:rand9.gz
        -out /etc/apache2/sites/www.midominio.com-ssl.key 1024
```

Eliminemos la palabra de paso introducida anteriormente

```
openssl rsa -in /etc/apache2/sites/www.midominio.com-ssl.key
        -out /etc/apache2/sites/www.midominio.com-ssl.pem
```

Generemos el Certificado de firma:

```
openssl req -new -key /etc/apache2/sites/www.midominio.com-ssl.pem
        -out /etc/apache2/sites/www.midominio.com-ssl.csr
```

Rellenemos los datos relativos a nuestro dominio:

```
openssl x509 -req -in /etc/apache2/sites/www.midominio.com-ssl.csr
        -signkey /etc/apache2/sites/www.midominio.com-ssl.pem
        -out /etc/apache2/sites/www.midominio.com-ssl.crt
```

[16] Para desactivarlo usaremos: `# a2dismod ssl`

Para autofirmar el certificado:

```
openssl x509 -req -in /etc/apache2/sites/www.midominio.com-ssl.csr
        -signkey /etc/apache2/sites/www.midominio.com-ssl.pem
        -out /etc/apache2/sites/www.midominio.com-ssl.crt
```

Podemos hacer un poco de limpieza:

```
rm -f /etc/apache2/sites/www.midominio.com-ssl.csr rand*.gz
```

Copiemos el fichero /usr/share/apache2/config/default-443 al lugar adecuado:

```
cp /usr/share/apache2/config/default-443 /etc/apache2/sitesavailable/
```

y ajustémoslo a nuestro sitio.

Creemos el enlace simbólico que permite activar este sitio:

```
ln -s /etc/apache2/sites-available/default-443 /etc/apache2/sites-enabled/default443
```

Reiniciemos Apache.

1.3. Loganizadores.

1.3.1. Definicición.

Se llama loganizadores a ciertos programas que analizan los archivos de logs del sistema para ofrecer estadísticas en un formato más entendible que el de dichos archivos.

1.3.2. Webalizer.

Se trata de un clásico en el mundo Debian, es un loganizador para Apache. Todo servidor web que se precie tiene que contar con un buen sistema de estadísticas que permita conocer el número de páginas que sirve, desde donde se accede a el, etc. todo ello lo conseguiremos con webalizer.

Se puede obtener en *ftp://ftp.mrunix.net/pub/webalizer/webalizer-2.01-10-src.tgz*.

También se tendrá que tener disponible la librería *gd-2.0.33.tar.gz* que se puede obtener en *http://www.boutell.com/gd*.

1.3.2.1. Instalación.

Se necesita instalar la librería gd[17] para el manejo de gráficos, lo realizaremos desde los ficheros fuentes de la librería siguiendo los siguientes pasos:

1. `tar zxvf gd-2.0.33.tar.gz`
2. `cd gd-2.0.33`
3. `./configure`
4. `make`
5. `make install`

Una vez concluida la instalación de la librería gd se está en disposición de instalar webalizer, antes de iniciarla se crearán dos directorios que la instalación utilizará:

1. `cd /usr/local`
2. `mkdir man`
3. `cd man`
4. `mkdir man1`

Y por fin, nos colocamos en el directorio donde está el fichero fuente y seguimos los siguientes pasos:

1. `tar zxvf webalizer-2.01-10-src.tgz`

[17] Es posible que se necesite instalar alguna librería más.

2. `cd webalizer-2.01-10`
3. `./configure --with-language=spanish`
4. `make`
5. `make install`

Al finalizar el proceso el programa ha quedado instalado en */usr/local/bin/webalizer* y un ejemplo de fichero de configuración en */etc/webalizer.conf.sample.*

1.3.2.2. Configuración.

Para empezar a configurar se copia el fichero de ejemplo al directorio DocumentRoot del servidor web con el nombre que se desee.

```
cd /var/www/html
cp /etc/webalizer.conf.sample estadis.conf
```

Se edita el fichero y se comienza a configurar:

LogFile	Define la ubicación del fichero de log que almacena los accesos al seridor web	LogFile /var/log/httpd/access.log
OutputDir	Indica el directorio donde se almacenarán las estadísticas generadas. Es importante que este bajo el directorio del servidor web para que las estadísticas puedan verse. Por ejemplo: /var/www/html/estadisticas se visualizará con http://localhost/estadisticas	OutputDir /var/www/html/estadisticas

Con estos cambios es suficiente para poder ver las estadísticas de acceso, ya sólo se necesita ejecutar webalizer para que las genere:

```
cd /var/www/html
/usr/local/bin/webalizer -c ./estadis.conf
```

1.3.2.3. Configurar el cron.

Cron es una utilidad de sistema que sirve para lanzar procesos con una periodicidad determinada, como por ejemplo copias de seguridad u otro tipo de procesos que deben ser lanzados de forma desatendida.

El paquete Cron provee dos utilidades, el demonio cron propiamente dicho y el editor de tareas, crontab, que es la herramienta que más nos interesa.

crontab -e es un script que lanza el editor vi y abre un fichero donde se almacenan todos los trabajos que se lanzan periódicamente. Este fichero tiene un formato específico que veremos a continuación; una vez editado, crontab se encarga de integrarlo en el sistema.

El formato de este fichero es el siguiente:

minutos horas dia mes diadelasemana comando

Minutos	Entre 0 y 59
Horas	Entre 0 y 23
Día	Entre 1 y 31
Mes	Entre 1 y 12
Dia de la semana	Entre 0 y 6. 0 es Domingo, 1 Lunes, ... 6 Sábado
Comando	El comando o comandos a ejecutar. Si no está en el PATH, hay que especificar toda su ruta.

Si ponemos un * se ejecutarán una vez por hora en el caso de las horas, y una vez por minuto en el caso de los minutos, ...

Ejemplo 1: Ejecutar todos los días un script de copia de seguridad a las 7:00 de la mañana:

*0 7 * * * /home/usuario/copiadeseguridad.sh*

Ejemplo 2: Ejecutar todos los primeros de mes un script de copia de seguridad a las 7:00 de la mañana:

*0 7 1 * * /home/usuario/copiadeseguridad.sh*

Ejemplo 3: ejecutar todos los viernes a las 21:30 un script de copia de seguridad:

*30 21 * * 5 /home/usuario/copiadeseguridad.sh*

Ejemplo 4: Ejecutar cada 15 minutos un script de copia de seguridad :

*0, 15, 30, 45 * * * * /home/usuario/copiadeseguridad.sh*
**/15 * * * * /home/usuario/copiadeseguridad.sh*

1.3.2.4. Automatizar la creación de estadísticas:

¿Y para qué todo esto? Pues cuando hemos ejecutado webalizer hemos generado las estadísticas en ese momento pero no se volverán a actualizar, lo deseable es que periódicamente se actualicen de forma desatendida, y ahí entra el cron.

Se ejecuta:

```
crontab -e
```

y se añade una línea para que se ejecute webalizar cada 15 mínutos.

**/15 * * * * /usr/local/bin/webalizer -c /var/www/html/estadis.conf*

Capítulo 2

Servidor FTP. Para qué sirve. ProFTPD: Instalación, configuración y administración.

2.1. Para qué sirve.

Un servidor FTP posibilita compartir recursos en una red, ubicando estos recursos en una estructura de directorios. Estos recursos son archivos. El acceso a esos archivos estará controlado por unas determinadas políticas de seguridad (acceso), y los usuarios podrán no sólo acceder a copiar estos archivos hacia su máquina local, sino que, si se les permite, podrán almacenar sus propios archivos en la estructura de directorios habilitada por el servidor. Los usuarios se conectan a un servidor FTP siguiendo un mecanismo similar a la conexión mediante HTTP. El usuario debe disponer en su máquina local de una aplicación que posibilite ser cliente de un servidor FTP. Normalmente los navegadores permiten conexión con servidores FTP, aunque puede recurrirse al uso específico de clientes FTP. Es preciso aclarar que un cliente FTP puede conectarse a cualquier servidor FTP (independientemente de qué programa servidor se esté empleando, nosotros lo haremos con ProFTPD, pero esto es extensible a cualquier otro), siempre que se hayan habilitado los permisos necesarios para que pueda llevar a cabo la conexión. En muchos casos, al habilitar las políticas de acceso a un servidor FTP, se incluye una configuración para permitir conexiones anónimas, es decir, sin precisar identificación de login y contraseña por parte del usuario. Hoy muchos servidores FTP existentes en Internet, y que ofrecen descargarse programas OpenSource, están configurados de esta manera. Incluso nosotros, cuando instalemos el servidor ProFTPD, podríamos optar por configurarlo de manera que cualquier usuario pueda acceder a los recursos almacenados sin precisar identificación explícita alguna.

Las siglas FTP hacen referencia a un protocolo de intercambio de datos denominado File Transfer Protocol. Los programas clientes y servidores se conectan e intercambian datos entre sí mediante este protocolo.

Los servidores FTP ofrecen una funcionalidad de gran potencia, debido a la sencillez ofrecida para obtener o depositar archivos, ya que la interfaz gráfica posee un aspecto similar a las aplicaciones de exploración de directorios existentes en los actuales sistemas operativos. Quizás como aspecto negativo está el hecho de que el usuario, al verse obligado a recorrer una estructura de directorios para localizar la información que necesita, está obligado a saber qué es lo que busca y a comprender la lógica de la estructura de directorios establecida. Por otra parte, es mediante FTP como habitualmente los usuarios pueden incorporar archivos en espacios de almacenamiento remotos de Internet. Por ejemplo, los proveedores de internet actuales ofrecen el servicio de FTP para que sus clientes puedan diseñar y colgar sus páginas webs personales en los espacios de almacenamiento contratados para tal efecto.

En el caso de un centro educativo, un servidor de FTP puede convertirse en una vía rápida para compartir archivos, o simplemente para acceder a información variada (por ejemplo, apuntes de clase, ejercicios, etc) sin necesidad de tener que diseñar una página web, o recurrir al correo electrónico. Además, la po-

sibilidad de controlar el acceso en base a nombres de usuario y contraseñas determinados abre grandes posibilidades de compartir gran cantidad de información en un sólo equipo para muchos usuarios diferentes.

Nosotros instalaremos un servidor FTP con el propósito de compartir datos tanto a nivel departamental como a nivel de grupos escolares. Además habilitaremos una zona para introducir recursos (software, manuales, etc) de acceso público.

Actualmente, todas las distribuciones de Linux incorporan servidores FTP. Aunque hay varios muy extendidos, recurriremos a ProFTPD para desarrollar el contenido de este apartado. Se trata de un servidor muy extendido, y básico en Red Hat y todas las distribuciones que se basan en ella. Posee un mecanismo de configuración parecido al seguido por el servidor web Apache, lo que lo hace aún más atractivo.

Llega el momento de plantearse qué utilidad podemos darle al servidor FTP en nuestro centro. Está claro que primero debe existir una necesidad de compartir información, o al menos, de ofrecerla para que otros la adquieran. Esta necesidad puede surgir en el mismo momento en que un profesor o un departamento decide hacer accesible a los alumnos los apuntes de una asignatura, enunciados de ejercicios e incluso sus soluciones. Si vamos un poco más lejos, puede convertirse en un medio para que los alumnos proporcionen sus trabajos a los profesores a través de la red, sin necesidad de recurrir al correo electrónico. Es más, podría incluso habilitarse para cada alumno un área en el servidor para que coloque sus trabajos y puedan éstos ser inspeccionados por sus profesores. Como se observa, el uso de un servidor FTP amplia y mejora la funcionalidad ofrecida por sistemas de compartición de recursos punto a punto como el ofrecido por los sistemas Windows de Microsoft.

Plantearemos por tanto un caso práctico, que posteriormente iremos ampliando. Por ahora la necesidad que surge es la de ofrecer una serie de recursos (programas de utilidad y documentacion variada) a toda la comunidad escolar. En principio planteamos un servidor de ámbito local, es decir, accesible a través de la red local del centro. Debe existir un directorio denominado Recursos, y dentro de éste se crearán otros dos: Software y Manuales.

Establecida la necesidad, procedemos a instalar el servidor ProFTPD.

2.2. Instalación.

¿Cómo obtener el servidor ProFTPD? Lo mejor es recurrir al paquete que proporcione la propia distribución Linux con la que estemos trabajando. Así nos aseguraremos que no habrá problemas de compatiblidad alguna con el resto de paquetes instalados. Otra opción es recurrir a internet para localizar paquetes de versión posterior al disponible en la distribución. Para ello puede recurrirse a http://rpmfind.net y comprobar si existen paquetes de versión posterior a la disponible en nuestra distribución (hay que asegurarse que está especialmente compilado para ella). Finalmente, la opción más aconsejable, aunque un poco más enredosa para aquellos que no están muy familiarizados con Linux, es compilar los archivos que componen el servidor desde sus propias fuentes. El que opte por esta última opción podrá descargarse la última versión disponible desde la propia página oficial de ProFTPD, que es http://www.proftpd.org.

Instalar el paquete disponible en la distribución

Escogemos esta opción, por ser la más asequible para todos. En el caso de Mandrake 10 (en Mandrake 9.1 la versión es la 1.2.7, y los paquetes RPMS son el proftpd-1.2.7-1mdk.i586.rpm y el proftpd-1.2.7-1mdk.i586.rpm), se precisan los siguientes paquetes:

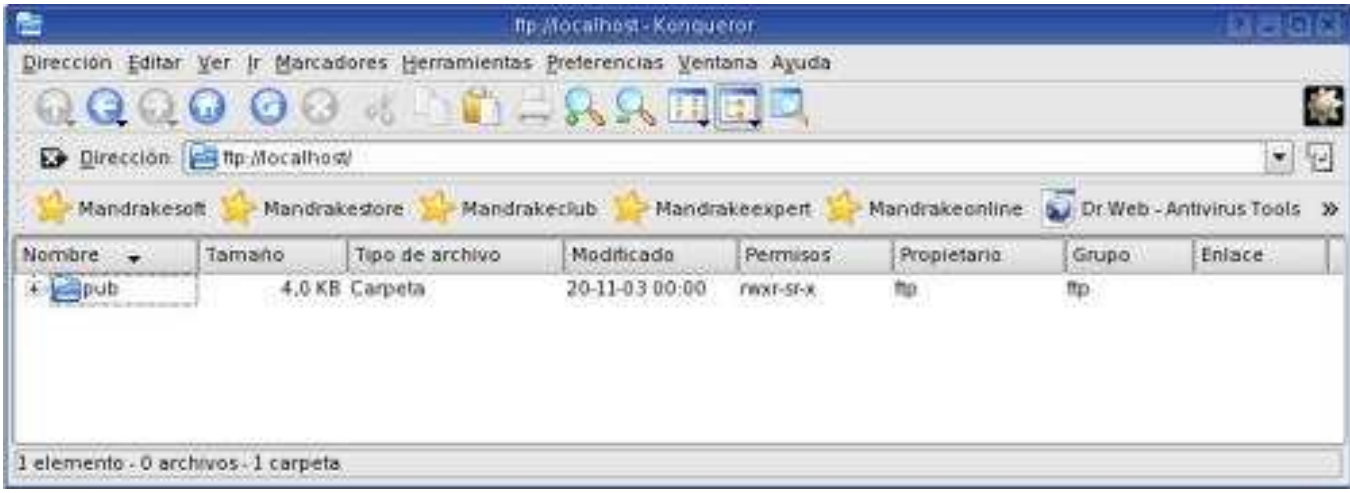
- **proftpd-1.2.9-3mdk.i586.rpm** (CD 1)
- **proftpd-anonymous-1.2.9-3mdk.i586.rpm** (CD 2)

Se puede recurrir tanto al comando rpm como a la herramienta gráfica de instalación de paquetes. Si disponemos de Guadalinex, deberemos tener el CD de suplementos. El servidor ProFTPD disponible en este CD se corresponde también con la versión 1.2.9 (aunque en este caso el formato no es rpm sino deb). Para instalarlo, bastará con introducir el CD de suplementos en la unidad lectora y tener Guadalinex 2004

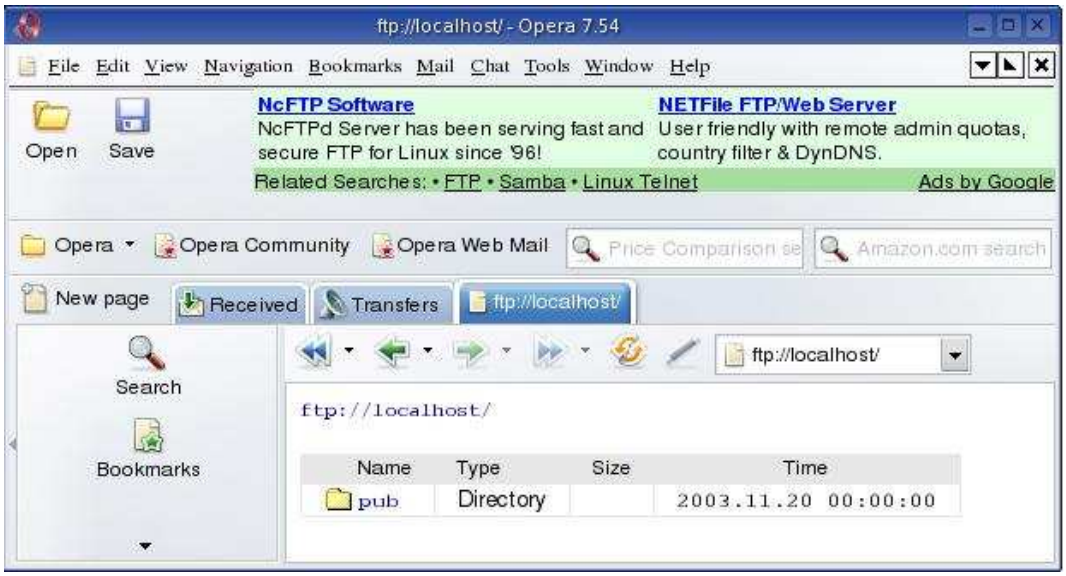
ya instalado y funcionando. En el menú Aplicaciones, opción Configuración->GMAX se iniciará el proceso para instalar el servidor (y cualquier otra aplicación del CD de suplementos).

Una vez instalado, podemos abrir el navegador e insertar como URL la siguiente: ftp://localhost. Entonces deberá aparecer algo similar a lo siguiente[1]:

- En Konqueror:

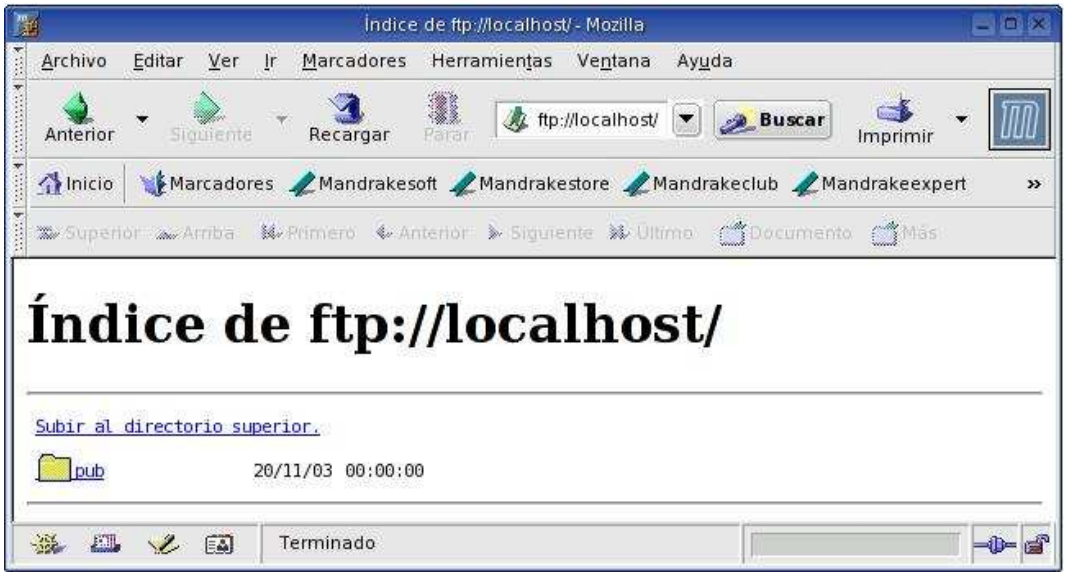


- En Opera:



- En Mozilla:

[1] Si trabajamos con Guadalinex 2004 no nos mostrará lo mismo, ya que no está habilitada la configuración para acceder en modo anónimo, que es como lo estamos haciendo en este instante. Más adelante se indicará la manera de solventar esto.



Aunque cada aplicación cliente (Konqueror, Opera, Mozilla) muestra la respuesta del servidor con ligeras diferencias, en todos queda claro que existe un directorio denominado **pub**.

Para confirmar que se está ejecutando, bastará con ejecutar el comando siguiente:

ps -ef | grep proftpd

Y la respuesta debe contener una línea similar a la siguiente:

nobody 3317 1 0 09:52 ? 00:00:00 proftpd: (accepting connections)

Otra forma de comprobar si está ejecutándose el servidor es mediante:

cat /var/run/proftpd.pid

Esta instrucción muestra el contenido del archivo proftpd.pid, que contiene el pid del proceso servidor.

Otra posibilidad es recurrir al script /etc/rc.d/init.d/proftpd[2]. Si ejecutamos:

/etc/rc.d/init.d/proftpd status

sabemos si se está ejecutando el servidor o no. En caso de que no se esté ejecutando se indicará que está detenido.

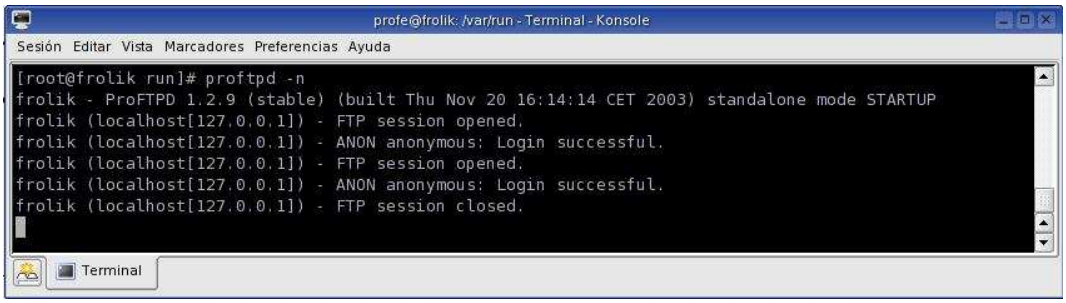
Finalmente, puede recurrirse al comando **service**[3] para comprobar si está corriendo el servidor o no, y además, para pararlo o arrancarlo. Para ello, si ejecutamos *service proftpd*, nos ofrecerá las posibilidades disponibles para gestionar proftpd mediante este comando:

I need an action
Uso: /etc/init.d/proftpd {start|stop|status|restart|reload|resume|suspend}
'suspend' acepta argumentos adicionales que se pasan a ftpshut(8)

Si ejecutamos *service -s | grep proftpd*, entonces sabremos si está detenido o corriendo, ya que aparecerá como respuesta un comentario al respecto.

Para arrancarlo bastará con ejecutar *service proftpd start* o bien simplemente, desde la terminal, ejecutar proftpd, siempre como usuario root. Si no queremos ejecutarlo como un proceso demonio, deberemos emplear la opción -n en el arranque. Para detenerlo ejecutaremos el comando kill seguido de su identificador de proceso. Por ejemplo, si lanzamos el servidor mediante *proftpd -n*, lo podemos parar consultando el pid en /var/run/proftpd.pid y ejecutando kill seguido del pid, o bien simplemente, haciendo service proftpd stop. Si tras detener el proceso, al ejecutar ps -ef | grep proftpd, nos aparece un comentario indicando que proftpd no existe pero que sí existe el archivo pid, bastará con borrarlo. El interés de ejecutar el servidor con el argumento -n es que en la consola veremos en cada momento qué conexiones se están procesando. Por ejemplo, en la siguiente imagen puede observarse que se han abierto dos sesiones y después se ha cerrado una de ellas.

[2] En Guadalinex 2004 el script está en /etc/init.d, sin embargo, la opción status no está implementada.
[3] No disponible en Guadalinex 2004.



Si no se empleara la opción -n, los mensajes producidos por el servidor se almacenarán en /var/log/syslog.

Finalmente, si al realizar la instalación se dispone del script /etc/rc.d/init.d/proftpd, bastará con ejecutar */etc/rc.d/init.d/proftpd start*. Para detenerlo, */etc/rc.d/init.d/proftpd stop* y para reiniciarlo (lo que supone releer el archivo de configuración), */etc/rc.d/init.d/proftpd restart* (consiste en realizar un stop y luego un start)[4].

Archivos que componen ProFTPD

El conjunto de archivos que intervienen en la ejecución, configuración y administración son los siguientes:

- /etc/proftpd.conf : Archivo principal de configuración del servidor.
- /etc/proftpd-anonymous.conf : Archivo de configuración del servidor para conexión anónima[5].
- /usr/sbin/proftpd : El ejecutable del servidor FTP.
- /etc/rc.d/init.d/proftpd : Un script que posibilita, entre otras cosas, arrancarlo y pararlo.
- /usr/bin/ftpwho : Muestra información de los procesos de todas las conexiones activas.
- /usr/bin/ftpcount : Muestra el número actual de conexiones del servidor.
- /usr/sbin/ftpshut : Permite detener el servidor, cerrando las conexiones abiertas y denegando las que posteriormente se soliciten.
- /var/run/proftpd.pid : Archivo que contiene el pid del proceso asociado al servidor.
- /var/log/xferlog : Archivo de registro en el que se almacenan las actividades de descarga y almacenamiento a través del servidor FTP.

2.3. Configuración.

Ahora que tenemos arrancado el servidor procederemos a analizar la estructura del archivo de configuración del servidor, /etc/proftpd.conf. Este archivo sigue una sintaxis similar al archivo de configuración del servidor Apache, por lo que no debería resultar complicado familiarizarse con él.

Al instalar ProFTPD, se creará un archivo de configuración por defecto. Este archivo tiene el contenido que seguidamente se muestra[6]:

 # This is a basic ProFTPD configuration file (rename it to
 # 'proftpd.conf' for actual use. It establishes a single server
 # and a single anonymous login. It assumes that you have a user/group
 # "nobody" and "ftp" for normal operation and anon.
ServerName "ProFTPD Default Installation"

[4] En Guadalinex 2004, /etc/init.d/proftpd start, /etc/init.d/proftpd stop, y /etc/init.d/proftpd restart.
[5] En Guadalinex 2004, el contenido de este archivo está integrado en proftpd.conf.
[6] En Guadalinex 2004 es ligeramente diferente. Esas diferencias se abordan a lo largo del texto.

ServerType standalone
DefaultServer on
 # Allow FTP resuming.
 # Remember to set to off if you have an incoming ftp for upload.
AllowStoreRestart on
 # Port 21 is the standard FTP port.
Port 21
 # Umask 022 is a good standard umask to prevent new dirs and files
 # from being group and world writable.
Umask 022
 # To prevent DoS attacks, set the maximum number of child processes
 # to 30. If you need to allow more than 30 concurrent connections
 # at once, simply increase this value. Note that this ONLY works
 # in standalone mode, in inetd mode you should use an inetd server
 # that allows you to limit maximum number of processes per service
 # (such as xinetd).
MaxInstances 30
 # Set the user and group under which the server will run.
User nobody
Group nogroup
 # To cause every FTP user to be "jailed" (chrooted) into their home
 # directory, uncomment this line.
#DefaultRoot ~
 # Normally, we want files to be overwriteable.
<Directory />
AllowOverwrite on
</Directory>
 # Needed for NIS.
PersistentPasswd off
 # Default root can be used to put users in a chroot environment.
 # As an example if you have a user foo and you want to put foo in /home/foo
 # chroot environment you would do this:
#
 # DefaultRoot /home/foo foo
Include /etc/proftpd-anonymous.conf

Como puede observarse, la última línea ejecuta un mandato que supone añadir a este archivo de configuración el contenido del archivo /etc/proftpd-anonymous.conf[7]. El contenido de éste último es el siguiente:

 # A basic anonymous configuration, no upload directories.
<Anonymous ~ftp>
User ftp
Group ftp
 # We want clients to be able to login with "anonymous" as well as "ftp"
UserAlias anonymous ftp
 # Limit the maximum number of anonymous logins
MaxClients 10
 # Don't make it require a valid password or shell.
RequireValidShell off
AnonRequirePassword off
 # We want 'welcome.msg' displayed at login, and '.message' displayed

[7] En Guadalinex 2004, el contenido de este archivo está integrado en proftpd.conf, por lo que no existe la directiva include. Esas líneas aparecen comentadas, y sería necesario eliminar esos comentarios (#) para que surtan efecto. Debe, de todas formas, dejarse comentadas las líneas siguientes que van desde <*Directory incoming*> hasta <*Directory*>. Antes de hacer ningún cambio en este archivo será necesario realizar una copia de seguridad del mismo, por si acaso ;). El directorio asignado en la configuración de acceso anónimo es /home/ftp.

 # in each newly chdired directory.
DisplayLogin welcome.msg
DisplayFirstChdir .message
 # Limit WRITE everywhere in the anonymous chroot
<Limit WRITE>
DenyAll
</Limit>
</Anonymous>

Este archivo contiene las directrices que seguirá el servidor para proporcionar acceso anónimo, es decir, sin requerir identificación alguna de los usuarios que soliciten conectarse a él. Estas directrices se establecen mediante *directivas* y *bloques de configuración*(que contienen directivas que se aplican exclusivamente al bloque). Las directivas van seguidas por un valor.

Comentemos algunas de las instrucciones más importantes contenidas en estos archivos.

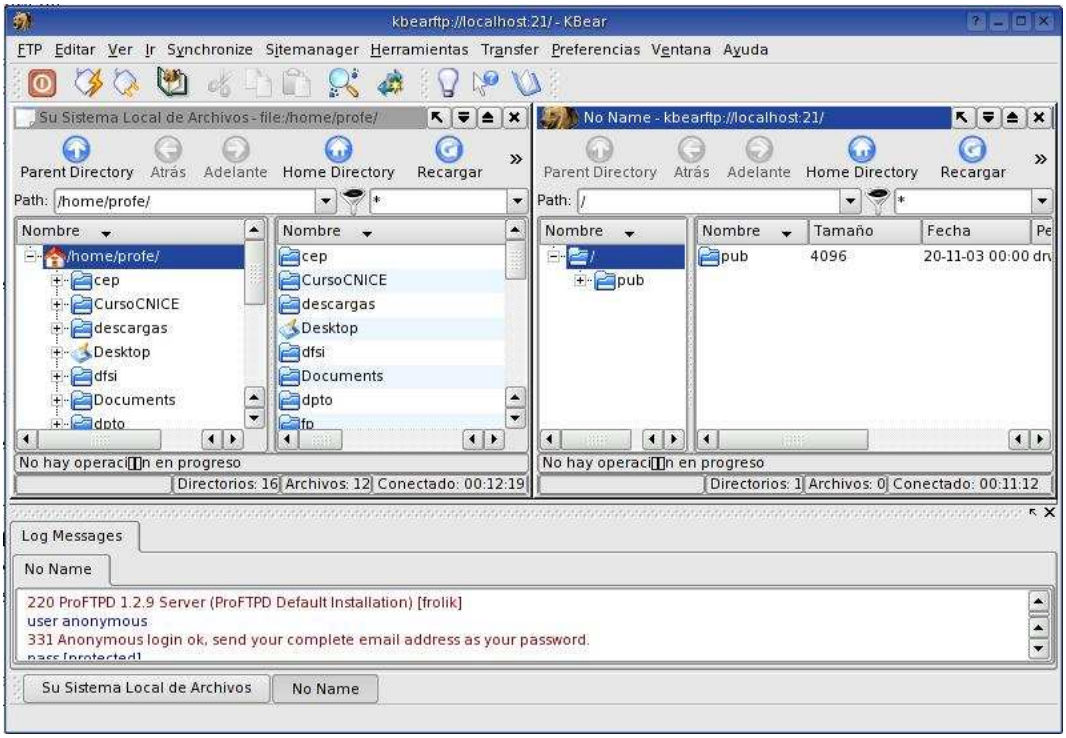
El comentario inicial del archivo /etc/proftpd.conf hace mención a la necesidad de que existan en el sistema dos usuarios concretos, **nobody** y **ftp**. No debemos preocuparnos, ya que la instalación basada en paquetes ya compilados genera estos usuarios si no existieran.

Para configurar el servidor bastará con editar /etc/proftpd.conf y /etc/proftpd-anonymous.conf y reiniciar el servidor. Otra posibilidad la tenemos con **webmin**. Se trata de una aplicación que mediante el protocolo http, y por medio del puerto 10000, permite configurar y administrar gran cantidad de componentes de un sistema Linux. Webmin está fuera del alcance de este curso, sin embargo baste decir que la mayoría de servidores pueden administrarse desde esta aplicación, así como acometer muchas operaciones habituales de administración en el sistema, como por ejemplo, la gestión de usuarios. Tiene, sin embargo, el inconveniente de que oculta al usuario qué archivos se están modificando y las acciones que lleva a cabo con cada uno de los componentes que se tocan. Consideramos que para conocer realmente cada uno de los servidores que se abordan en este módulo, se hace imprescindible gestionarlos desde las herramientas que se proporcionan con ellos.

Directivas y bloques de configuración más importantes

Directiva ServerName

Esta directiva establece la cadena de texto que se mostrará a los usuarios que se conecten al servidor. Este mensaje se visualizará si se está empleando un cliente específico de FTP (no es el caso de los navegadores). Cuando se emplea un cliente de FTP (por ejemplo, Mandrake proporciona Kbear), siempre se muestra un área en la que se visualiza el *diálogo* mantenido entre nuestro cliente y el servidor. A título de muestra, la siguiente imagen, que es una captura de pantalla de Kbear, permite comprobar que al conectarnos con el servidor (en este caso, al hacerlo sobre la propia máquina, localhost), éste lo primero que hace es devolver al cliente la cadena correspondiente a **ServerName,** seguida del nombre de la máquina. El uso de un cliente de FTP se abordará un poco más adelante. Por ahora bastará con saber que en la actualidad, la mayoría de clientes FTP se ejecutan en modo gráfico, ofreciendo siempre, al menos, tres áreas: una ventana que proporciona acceso al sistema de ficheros local, otra que proporciona acceso al sistema de ficheros que ofrece el servidor y otra con los mensajes que se intercambian cliente y servidor. En el caso de esta captura de pantalla, el área inferior es la que muestra ese intercambio de mensajes. En ella podremos saber en cada momento qué operaciones estamos realizando sobre el servidor y cómo responde éste a ellas.



Directiva ServerType

Establece el modo de operación del servidor. Existen dos modos de operación: inetd y standalone. Por defecto es standalone. Si fuera inetd, el demonio se ejecutaría a través de otro proceso que es inetd (o xinetd), el cual podríamos denominar como superservidor, ya que controla la ejecución de varios servidores. El proceso xinetd posibilita arrancar programas que proporcionan ciertos servicios dentro del marco de Internet, como puede ser un servidor web, un servidor de correo o el servidor FTP. Cuando se ejecuta como standalone, lo que se consigue es que el servidor quede lanzado y a expensas de recibir peticiones por el puerto correspondiente, es decir, no está controlado por ningún otro proceso. El uso de inetd (xinetd) está fuera del alcanze de este curso.

Directiva DefaultServer

Se pueden configurar varios servidores *virtuales* sobre un único servidor FTP. En este caso, cualquier conexión que no especifique el servidor virtual, se asociará con un servidor virtual por defecto. Como en este momento sólo hay un servidor, esta directiva la dejaremos a **on** en este punto del archivo de configuración.[8]

Directiva Port

Establece el puerto por el que atenderá las peticiones el servidor. Los puertos en un ordenador se pueden considerar como puntos de entrada y salida de datos entre una aplicación que se ejecuta en la máquina local y otra que está es un ordenador remoto, aunque también esto puede ser exensible a aplicaciones que se ejecutan en una misma máquina. Los puertos del 1 al 1024 están reservados para el sistema, y del 1025 hasta el 65535, para otras aplicaciones. En el caso de FTP, el sistema reserva dos puertos, el 20 y el 21 (este último es el que está disponible públicamente para realizar la transferencia de archivos por FTP). Esto podemos cambiarlo, y optar por un puerto superior al 1024 para el servidor de FTP. Sin embargo, se recomienda dejar la asignación por defecto, ya que de esta forma, cada vez que se haga una petición de conexión al servidor, el usuario no estará obligado a realizar referencia alguna al puerto (ya que el 21 es el

[8] Más adelante se abordará el uso de servidores virtuales.

tomado por defecto siempre que se hace uso del protocolo de FTP). Si estableciéramos un puerto diferente, por ejemplo el 1050, entonces, cada vez que se requiera conectar con el servidor, será necesario añadir a la url del servidor, el puerto 1050. Por ejemplo, si disponemos de un servidor ftp en nuestra máquina local, y se le ha asignado el puerto 1050, la url en nuestro navegador será la siguiente:

ftp://localhost:1050

Observamos que la url comienza con la palabra ftp, haciendo referencia al protocolo que se debe emplear para gestionar la petición introducida.

En el archivo de configuración, el valor por defecto es 21. Finalmente, advertir que esta directiva tiene sólo efecto cuando se ejecuta el servidor en modo *standalone*, ya que para el caso de *inetd (xinetd)*, será el archivo de configuración de éste quien determine el puerto[9].

Directiva MaxInstances

Determina el número máximo de conexiones concurrentes que puede soportar en un instante determinado el servidor. Por defecto está puesto a 30, lo que significa que no se podrán servir más de 30 conexiones a la vez (imaginemos que el servidor puede ser accedido desde cualquier punto de Internet). Cuando llegue la 31, ésta no se aceptará. Esto evita que pueda llegar a sobrecargarse el servidor con excesivas peticiones.

Directiva DefaultRoot

Indica el directorio raíz desde el que el usuario que se conecte podrá navegar, siempre en profundidad, nunca más arriba de este directorio raíz. Si determinamos como directorio raíz lo siguiente:

DefaultRoot ~

Estaremos indicando que el directorio raíz será el directorio *home* del usuario (si se conecta un usuario con el login *alumno*, el directorio raíz será /home/alumno). De esta forma se puede asignar a cada usuario un área específica de ftp para él, en la que no podrán entrar el resto de usuarios. Por defecto viene esta directiva comentada (precedida para ello por #). Esto significa que cualquier usuario que se conecte quedará ubicado en el directorio raíz por defecto del servidor. En el caso que nos ocupa, será /var/ftp/pub.

Bloque de configuración Directory

Este bloque posibilita establecer un conjunto de directivas que se aplican exclusivamente sobre un determinado directorio (y sus correspondientes subdirectorios si existieran). El siguiente fragmento:

<Directory />
AllowOverwrite on
</Directory>

especifica que para el directorio raíz del servidor (en nuestro caso, /var/ftp/pub) se aplicará la directiva *AllowOverwrite on*. El bloque viene delimitado por la pareja <Directory></Directory>(al estilo HTML).

Directiva AllowOverwrite

Permite sobreescribir archivos si se establece su valor a on.

Bloque de configuración Anonymous

Especifica un conjunto de directivas que se aplican a los accesos anónimos. Este bloque de configuración va acompañado del directorio raíz en el que se ubicará el usuario anónimo cuando se conecte al servidor. Por defecto es /var/ftp/pub. Pero si se desea, por ejemplo, asignarle el directorio /home/publico, entonces se escribirá *<Anonymous /home/publico>*. El ámbito del bloque viene determinado por la pareja *<Anonymous></Anonymous>*.

[9] Se recomienda hacer un *man inetd* o *man xinetd* si se desea obtener más información, así como consultar el archivo */etc/services* para conocer los puertos y procesos (servicios de internet) que controlará este superservidor.

Directiva UserAlias

Posibilita crear alias de usuarios del sistema. Un ejemplo práctico es precisamente con el acceso anónimo al servidor. En este caso, el login que se precisa es ftp (usuario ya existente en el sistema). Sin embargo, en muchas ocasiones, si se debe proporcionar una identificación de usuario anónimo, se prefiere emplear como login la palabra *anonymous*. Sin utilizar un alias, nos veríamos obligados a crear una nueva cuenta de usuario en el sistema, y asignarle como grupo el mismo al que pertenece ftp (el grupo es ftp). Esta es la razón por la cual se emplea la directiva UserAlias dentro del bloque de directivas para la conexión anónima con el propósito de disponer de este *anonymous* como alias de *ftp*.

Directiva MaxClients

Número máximo de clientes anónimos. Por defecto es 10. Puede no establecerse límite empleando el argumento *none*. Ejemplo: *MaxClients none*. También puede acompañarse de un mensaje que se ofrecerá al usuario que pretende conectarse y que es rechazado por haberse alcanzado el límite de conexiones: *MaxClients 10 "Se ha alcanzado el máximo de conexiones anónimas disponibles [%m]"*, donde *%m* será sustituída por 10.

Directiva DisplayLogin

Muestra un mensaje cuando un usuario se conecta. En el caso de un usuario anónimo, se muestra el contenido del archivo welcome.msg. Este archivo, si no se indica camino absoluto alguno, se entenderá que está ubicado en el directorio raíz. Puede establecerse cualquier otro archivo, por ejemplo: *DisplayLogin saludo.txt*.

Directiva RequireValidShell

Permite establecer si el usuario que se conecte al servidor debe poseer una shell válida o no. En el caso de usuarios anónimos (que no están registrados como tales en la máquina donde se ejecuta el servidor) se hace conveniente dejarla a *off*.

Directiva AnonRequirePassword

De forma similar a la anterior, esta directiva obligará o no a que el usuario introduzca una password o no. La password para usuarios anónimos no tiene demasiado sentido, y en todo caso, lo que se hace es solicitar al usuario anónimo como password su dirección de correo electrónico. Si establecemos esta directiva a *off*, el usuario podrá decidir si desea o no introducir algo como password. Si no introduce nada, se aceptará esta conexión de todas formas.

Bloque de configuración Limit

Establece restricciones sobre determinados comandos o acciones que pueden realizarse sobre el servidor. Por ejemplo,

<Limit WRITE>
DenyAll
</Limit>

Establece que no se permite la escritura dentro del bloque en el que esté incluido. En el caso que nos ocupa, está incluido en el bloque de directivas asociadas al acceso anónimo, de forma que no se permitirá a ninguna conexión anónima escribir en la jerarquía de directorios proporcionada por el servidor.

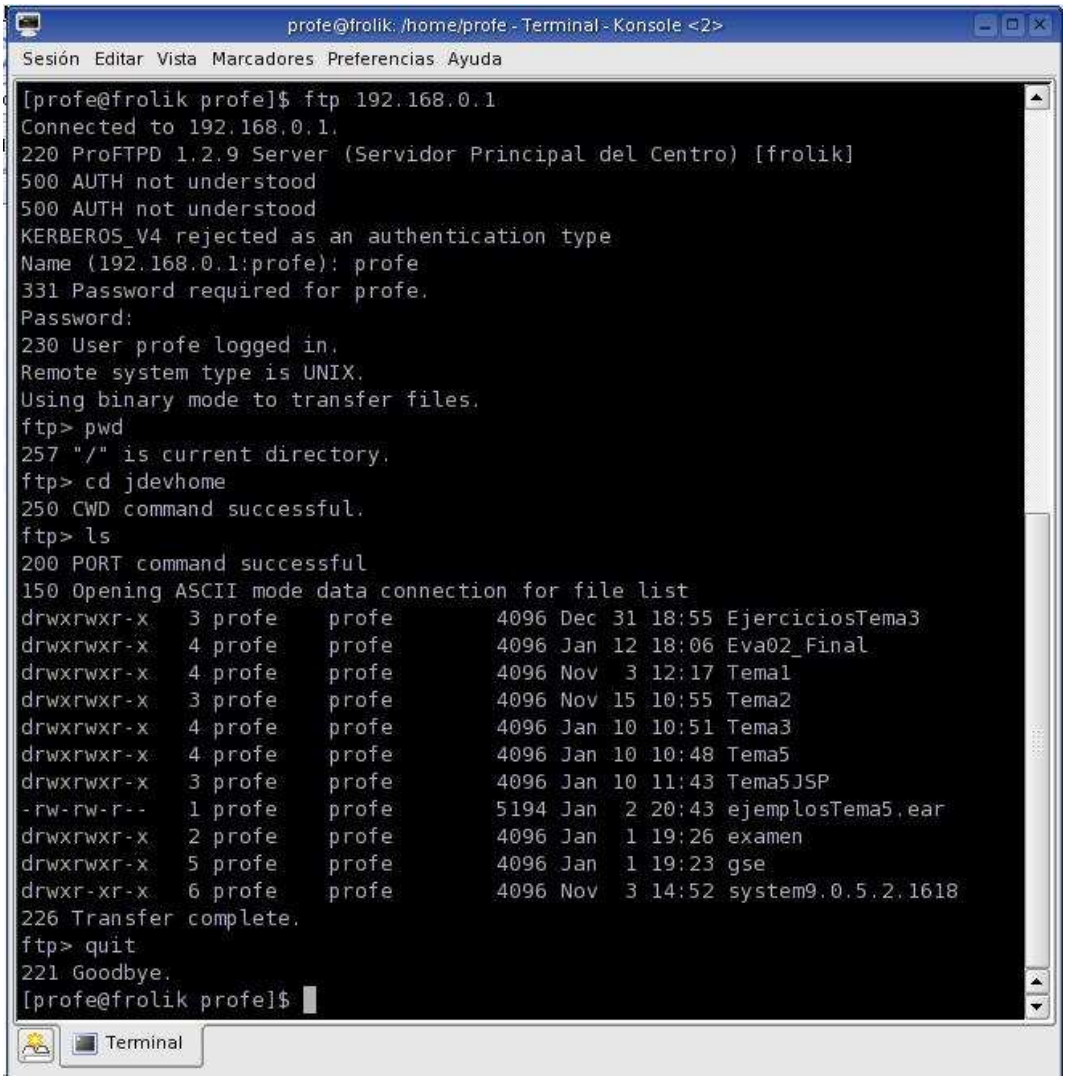
Con esto concluímos un breve repaso por las directivas y bloques de configuración más importantes del servidor.

Ha llegado el momento de empezar a sacarle partido, pero antes, un breve apunte sobre los clientes ftp, tanto en modo texto como en modo gráfico.

Clientes de FTP.

Para poder acceder a los recursos disponibles en un servidor de FTP se hace necesario ejecutar algún programa que, mediante el protocolo FTP, establezca el diálogo oportuno con el servidor para realizar las operaciones necesarias sobre él (copiar archivos en él, copiarlos desde el servidor hacia la máquina del usuario, navegar por la jerarquía de directorios del servidor, borrar archivos y carpetas, crearlas, etc). Estos programas reciben el nombre de clientes FTP. Pueden ser programas que se ejecutan en modo texto sobre un terminal linuz (consola), o bien ser aplicaciones gráficas. La diversidad de clientes FTP es bastante elevada (hay para todos los gustos y colores). Sin embargo los siguientes son los más utilizados:

- **Cliente ftp en modo texto, integrado con Linux**. Este comando está disponible en cualquier versión/distribución de Linux. Tratar este comando puede suponer elaborar todo un manual para él solito, aunque, sin embargo, si se consulta *man ftp* se podrá obtener una visión general de su funcionamiento. En la siguiente captura de pantalla se observa una sesión de ftp en modo texto:



Y en la siguiente dos procesos de transferencia, uno de descarga y otro de subida.

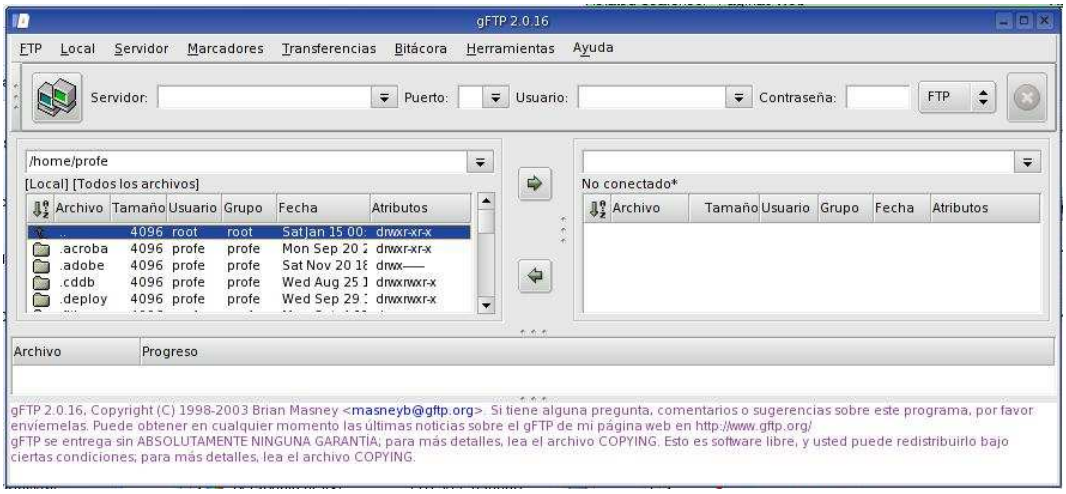
```
ftp> put listadeprogramas.txt
local: listadeprogramas.txt remote: listadeprogramas.txt
200 PORT command successful
150 Opening BINARY mode data connection for listadeprogramas.txt
226 Transfer complete.
ftp> get ejemplosTema5.ear
local: ejemplosTema5.ear remote: ejemplosTema5.ear
200 PORT command successful
150 Opening BINARY mode data connection for ejemplosTema5.ear (5194 bytes)
226 Transfer complete.
5194 bytes received in 0.00064 seconds (8e+03 Kbytes/s)
```

- **gFTP**. Un cliente FTP que cuenta con un interface gráfico muy intuitivo. Permite descargar ficheros arrastrándolos y soltándolos en distintas ventanas y transferir directorios completos. Soporta los protocolos de transferencia de ficheros más comunes: FTP, HTTP y SSH. Por medio de las opciones de configuración permite hacer conexiones a través de un servidor Proxy. Es ideal para las personas que no tienen práctica con los clientes de ftp por medio de comandos. Viene incluido tanto en Mandrake como en Guadalinex[10].
- **KBear**. Cliente FTP gráfico para KDE. Se instala por defecto y es similar a gFTP[11].
- **Usar un navegador para acceder al servidor**. Esta puede ser la vía más idónea cuando lo único que deseamos es *descargar* archivos desde el servidor. Sin embargo, si pretendemos realizar tareas de *subida* de archivos, creación de directorios, borrado, etc., entonces se hace necesario usar un cliente específico de FTP.

Introducción a gFTP

Para instalarlo basta con proceder como lo hemos hecho con paquetes anteriores, recurriendo a apt-get (caso de distribuciones Debian como Guadalinex) o con rpm (para distribuciones Red Hat y derivadas, como el caso de Mandrake). Si trabajamos con Guadalinex, gFTP ya lo tendremos instalado. En caso de Mandrake, si durante la instalación no incluímos este cliente, bastará con recurrir a rpm para instalarlo. También podemos hacerlo desde el entorno gráfico recurriendo al gestor de paquetes gráfico.

Al ejecutar gFTP, nos aparece una ventana como la siguiente:

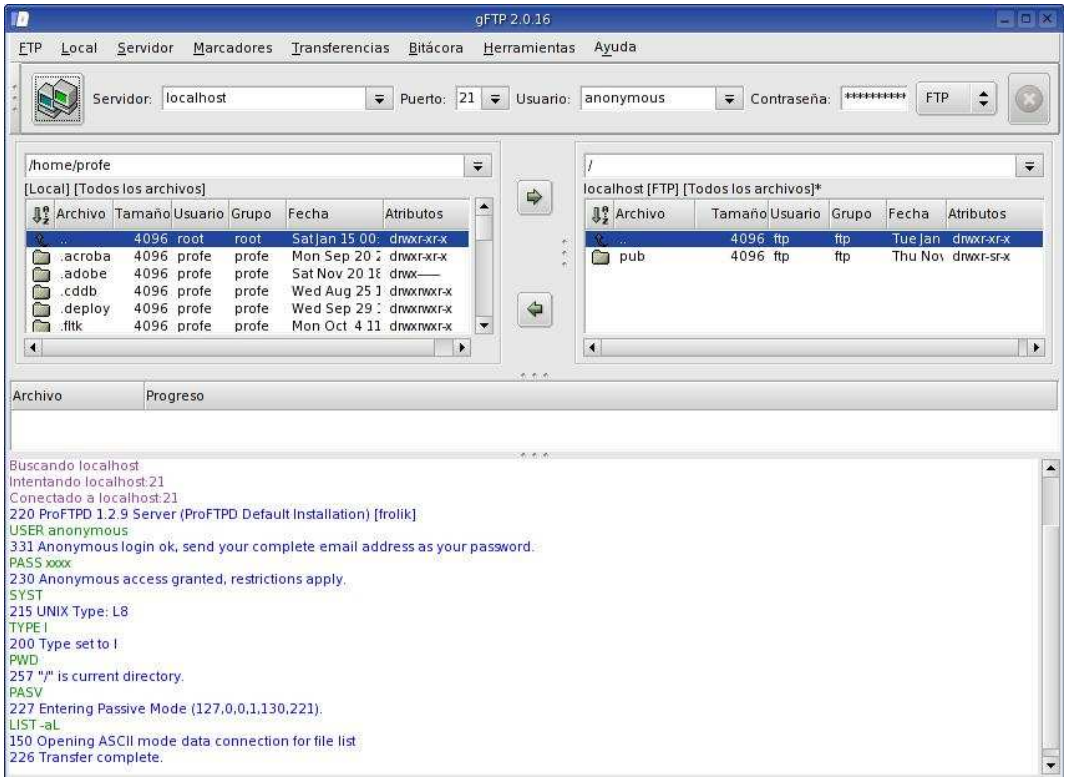


Si observamos, bajo la línea de menú principal se muestran una serie de cuadros de texto en los que se deberán introducir los datos correspondientes al servidor, puerto de escucha, login de usuario y contraseña.

[10] Existe, tanto en Mandrake como en Guadalinex 2004 un cliente ftp derivado de gFTP, se trata de gftp-text, y como puede intuirse, trabaja en modo texto.

[11] Disponible tanto en Mandrake como en Guadalinex 2004.

Si no rellenamos el campo correspondiente al puerto, se tomará por defecto el 21. Si tampoco rellenamos usuario y contraseña, se entenderá que se desea realizar una conexión anónima. Si introducimos *localhost* en el área destinada al servidor y luego pulsamos el icono  , entonces gFTP mostrará lo siguiente:



Resulta interesante inspeccionar el intercambio de mensajes entre el servidor y el cliente (área inferior). Los textos en azul corresponden al servidor y los de color verde al cliente. Además, disponemos en la zona izquierda de la ventana del programa un cuadro con la estructura de directorios de la máquina local y en la derecha la ofrecida por el servidor. Como no hemos modificado nada del archivo de configuración, el directorio del servidor ftp es /var/ftp/pub (el directorio raíz es /var/ftp en Mandrake y /home/ftp en Guadalinex, aunque en el cliente aparece como /). Ahora, si hacemos doble clic sobre *pub* accederemos a su interior. La forma de manejarse por la estructura de directorio es similar a como se hace con cualquier explorador de archivos convencional. Por supuesto, todas estas acciones tienen su correspondiente intercambio de mensajes en la zona inferior. Esta puede ser una buena forma de empezar a familiarizarnos con los comandos que se emplean para el intercambio de archivos con FTP.

Si transcurre un cierto tiempo sin realizar operaciones sobre el servidor, éste nos desconectará. Por defecto el tiempo establecido es de 300 milisegundos (se establece mediante la directiva *TimeoutNotTransfer*).

Ahora procederemos de manera que nos conectaremos al servidor mediante una cuenta ya existente en el sistema. Supongamos que la cuenta es *alumno*, y su contraseña también es *alumno*. Entonces, al conectarnos el servidor nos ubicará en el directorio */home/alumno*. Hay que recordar que, aunque deja al usuario en su directorio home, éste puede *ascender* por la jerarquía de directorios sin problema alguno. Es decir, puede moverse por toda la máquina donde está corriendo el servidor. Esto puede provocar problemas, ya que nunca, al montar un servidor FTP, pretendemos que los usuarios tengan acceso completo a la máquina. Nuestro objetivo será confinar siempre a los usuarios que se conecten a ciertos directorios, desde los cuales, y siempre hacia *abajo*, podrán recorrerlos. Llegados a este punto debemos plantearnos qué política deseamos seguir, si confinar a cada usuario en su directorio *home*, o ubicarlos a todos en uno solo. Esto va a

depender de para qué montamos el servidor de FTP. Si deseamos diseñar un *repositorio* de información que podrán descargarse los usuarios (y en todo caso, algunos de ellos con privilegios para almacenar archivos), entonces lo lógico es crear una estructura de directorios común a todos ellos. La otra posibilidad es disponer del servidor para que los usuarios posean un área de almacenamiento en esa máquina para depositar sus archivos, de forma aislada respecto a los demás. Finalmente, existe una tercera política consistente en mezclar las dos anteriores.

2.4. Administración

Retomemos el caso práctico que nos creó la necesidad de instalar un servidor FTP. En el centro educativo deseamos realizar los siguiente:

- Crear un área pública, para que los usuarios puedan descargarse programas de utilidad y educativos y manuales o tutoriales.
- Crear áreas de acceso exclusivo para determinados departamentos, que desean que sus integrantes puedan intercambiar información a través de ellas. Estas zonas deben permitir tanto la lectura como la escritura de archivos y carpetas.
- Posibilitar que cada profesor del centro disponga de un área reservada donde colocar sus archivos.
- Crear áreas para determinados grupos escolares, en las cuales podrán acceder en modo de escritura sus integrantes, aunque serán públicas en modo sólo lectura para el resto de la comunidad educativa.

Creación del área pública

Decidimos que se va a colocar en /ftp/public. Para ello primero creamos el directorio /ftp/public, como root. Seguidamente, hacemos una copia de seguridad de los archivos de configuración (/etc/proftpd.conf y /etc/proftpd-anonymous.conf) y editamos el archivo /etc/proftpd-anonymous.conf y lo modificamos de la siguiente manera:

donde aparece <*Anonymous ~ftp*> lo sustituímos por <*Anonymous /ftp/public*>[12].

Reiniciamos el servidor, por ejemplo ejecutando el script /etc/rc.d/init.d/proftpd restart[13] (no olvidar realizar esta operación como usuario root).

Al conectarnos con gFTP como usuario anónimo, observaremos que dentro del directorio raíz del servidor no aparece contenido alguno. Eso es debido, lógicamente, a que no se ha almacenado aún ningún archivo en su interior. Volvamos al explorador de archivos de Linux, o bien a un terminal de texto, y añadamos la siguiente estructura de directorios:

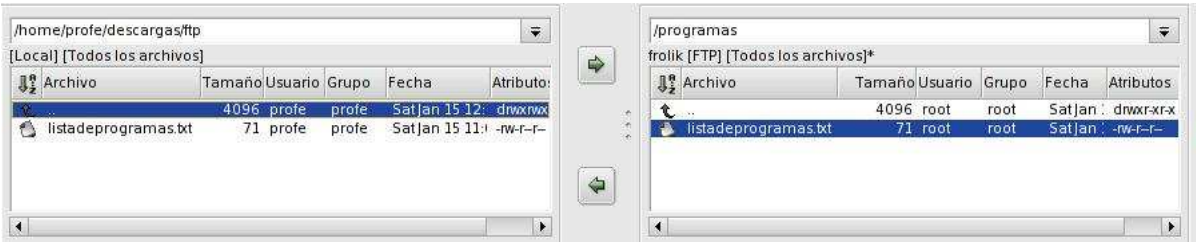
En el directorio manuales almacenaremos manuales y tutoriales, bastará con que copiemos en él un archivo cualquiera, incluso .txt, se trata sólo de realizar un ejemplo. En el directorio programas procederemos de forma similar. Realizadas estas operaciones nos conectamos de nuevo como usuario anónimo, y comprobaremos que ya tenemos acceso a dos directorios y sus correspondientes archivos.

Para copiar un archivo desde el servidor hacia nuestra máquina local (a esto lo denominaremos desde ahora *descargar* un archivo), bastará primero que seleccionemos el directorio local donde lo queremos

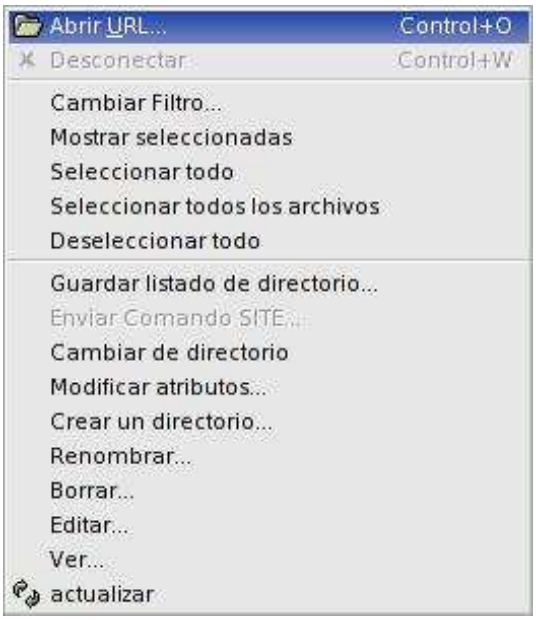
[12] En caso de usar Guadalinex, bastará con eliminar los comentarios que están aplicados a las directivas comprendidas entre <Anonymous>y </Anonymous>de /etc/proftpd.conf. Y después, realizar la sustitución que se indica.

[13] Recordar en caso de Guadalinex: /etc/init.d/proftpd restart

depositar (explorador de archivos de la máquina local), y luego haremos doble clic sobre el archivo a descargar. Si queremos descargar el archivo /manuales/listademanuales.txt, haremos doble clic con el ratón sobre él y lo tendremos copiado en el directorio local seleccionado.



Hemos copiado listadeprogramas.txt desde /programas (recordemos que el directorio raíz para los usuarios es /ftp/public, y se representa por /) hacia /home/profe/descargas/ftp. Si hacemos clic con el botón derecho del ratón en el explorador de archivos local, podremos advertir las operaciones que se pueden realizar sobre él (entre otras, crear archivos, directorios, borrarlos, etc.).



Estas mismas opciones nos aparecen en el explorador de archivos del lado del servidor, sin embargo, la mayoría de las opciones no las tendremos disponibles como usuario anónimo. Basta con intentar crear un directorio para que en la ventana de mensajes nos aparezca algo similar a esto:

En resumen, como usuario anónimo sólo podemos recorrer el árbol de directorios y descargar archivos.

Procedamos ahora como un usuario registrado. Al conectarnos como usuario *alumno* observamos que el servidor nos deposita en */home/alumno*, y que nos podemos mover libremente hacia arriba y hacia abajo del árbol de directorios. Por supuesto, podemos llegar a /ftp/public. En este caso, las operaciones disponibles para el usuario alumno dependerán de los permisos asignados a directorios y archivos. Esta situación puede no ser recomendable. Como el directorio /ftp/public es público y de solo lectura para todos, podemos determinar que si un usuario registrado desea acceder a él, lo haga como usuario anónimo. Y que si accede como usuario registrado, se le ubique en su directorio *home*, pero sin posibilidad de ascender a partir de

él. Para ello editamos el archivo /etc/proftpd.conf, y le quitamos el comentario (#) a la directiva *DefaultRoot ~*. Guardamos el archivo y reiniciamos el servidor. Ahora el directorio raíz mostrado por el servidor corresponde al directorio home del usuario.

Crear áreas reservadas para cada profesor

Crear estas áreas ya es automático, basta con que cada profesor disponga de una cuenta de usuario en la máquina servidora. Así si un profesor tiene como cuenta *prof1*, al conectarse al servidor quedará ubicado en su directorio */home/prof1*. Es conveniente recordar que los directorios *home* los podemos especificar nosotros, es decir, que una cuenta puede tener asociado un directorio *home* diferente al asignado por defecto por el sistema.

Crear áreas exclusivas para departamentos

Supongamos que queremos empezar por asignar un área de acceso exclusivo para los profesores que pertenecen al departamento de la Familia Profesional de Informática. Hay que empezar por establecer su estructura de directorio. En nuestro caso puede ser */ftp/dpto/informatica*. Ahora debemos configurar el servidor para que todos los que pertenecen al departamento entren en el área exclusiva para ellos. Aunque esto puede resolverse de varias formas, una de las más fáciles es crear un usuario en el sistema que se llame por ejemplo *dptoinfo*. Este usuario tendré una contraseña sólo conocida por los integrantes del departamento y su directorio *home* tendrá que establecerse a */ftp/dpto/informatica*.

Crear áreas para grupos, de acceso completo para los usuarios registrados en ella, y limitado a lectura para el resto

Comencemos por crear */ftp/grupos/eso*. El acceso a este área puede realizarse de forma similar a cómo se hizo con el área del departamento. Se crea un usuario y grupo determinados, y se asocia este directorio. Esta vez procederemos de otra manera. Establecemos las siguientes directivas:

DefaultRoot ~ !alumno
DefaultRoot /ftp/grupos/eso alumno

De esta manera estamos haciendo lo siguiente:

- Asociamos como directorio raíz el home de los usuarios que se conecten, salvo a los que pertenecen al grupo alumno.
- Establecemos */ftp/grupos/eso* como directorio raíz para el grupo *alumno.*

Por supuesto, en el sistema, cada cuenta de usuario perteneciente al grupo alumno tendrá diferentes directorios home, pero éstos no son considerados en la configuración del servidor. Esto ofrece una mayor flexibilidad, sobre todo cuando los directorios home de los usuarios registrados no deben modificarse.

Ya lo que queda es permitir acceso anónimo a */ftp/grupos/eso*. Esta vez recurriremos al uso de un servidor virtual. Un servidor virtual posibilita trabajar con una estructura de directorios y control de acceso alternativos al servidor principal, y todo ello dentro del mismo archivo de configuración y gestionado por el propio ProFTPD. La idea en el caso concreto que nos ocupa es creador otro servidor FTP dedicado exclusivamente a servir a usuarios anónimos el contenido del directorio /ftp/grupos/eso. Para ello se introducen las siguientes directivas inmediatamente antes de la línea final de include:

```
<VirtualHost 192.168.0.1>
  ServerName "Servidor Virtual. Acceso público a grupos ESO"
  DefaultRoot /ftp/grupos/eso
  Port 2121
<Limit LOGIN>
  DenyAll
</Limit>
<Anonymous /ftp/grupos/eso>
```

User ftp
Group ftp
RequireValidShell off
AnonRequirePassword off
UserAlias anonymous ftp
<Limit LOGIN>
AllowAll
</Limit>
</Anonymous>
</VirtualHost>

Estas directivas realizan lo siguiente:

- Se establece un servidor ftp virtual asociado a la IP 192.168.0.1. Podríamos haber hecho referencia al nombre de la máquina. Realmente, en este caso, esta IP (o el nombre de la máquina) son los mismos que se toman para el servidor principal (por lo que huelga decir que la IP del equipo donde está ubicado el servidor es 192.168.0.1[14]). Sin embargo no va a existir conflictos entre un servidor (el principal) y el otro (el virtual), ya que el primero atiende peticiones por el puerto 21, mientras el segundo lo hace por el 2121. Para ello se emplea la directiva Port. Además, se establece que el directorio raíz será */ftp/grupos/eso*. Y además, se establece que ningún usuario podrá conectarse al servidor.

- Si ningún usuario puede conectarse al servidor, ¿qué sentido tiene crearlo? Pues fácil, ahora especificamos cómo será el funcionamiento de éste cuando se intente conectar algún usuario de forma anónima. Para ello recurrimos a la directiva <Anonymous>. Determinamos que el servidor atenderá peticiones anónimas en */ftp/grupos/eso*, que no requerirán una shell válida ni password, con las directivas User y Group determinamos el usuario y grupo que será aceptado por la conexión anónima, y que podrá identificarse tanto como *anonymous* como *ftp*. Además, ahora establecemos con la directiva <Limit>, que se permitirá la conexión a todos los usuarios permitidos (que es el usuario *ftp* del grupo *ftp*, y su alias, *anonymous*).

Resumiendo, en este momento tenemos el siguiente proftpd.conf:

#Archivo de configuración para un servidor ProFTPD de un centro educativo
ServerName "Servidor Principal del Centro"
ServerType standalone
DefaultServer on
 # Permite restaurar transferencias que se han cortado desde un cliente
 # hacia el servidor
AllowStoreRestart on
 # Puerto estándar de FTP
Port 21
 # Máscara para los permisos que se asociarán a los archivos y
 # directorios que se creen en el servidor
Umask 022
 # Número máximo de conexiones concurrentes
MaxInstances 30
 # Usuario y grupo bajo el que se ejecutará el servidor.
User nobody
Group nogroup
 # Todos los usuarios quedarán ubicados en su directorio home, salvo
 # los pertenecientes al grupo alumno.
DefaultRoot ~ !alumno

[14] También se podría haber optado por crear un alias para la interfaz eth0 si no disponemos más que de una y deseamos disponer de direcciones IP diferentes para cada uno de los servidores. En ese caso, tras crear el alias, podríamos asociar el servidor virtual a la segunda IP creada (por ejemplo, eth0 puede tener la IP 192.168.0.1, y eth0:0 la IP 192.168.0.2).

```
# Se permite la sobreescritura de archivos en el directorio raíz.
<Directory />
  AllowOverwrite on
</Directory>
# Para el directorio /ftp/grupos/eso, se permite subir archivos a los
# usuarios del grupo alumno, y al resto no. Primero se aplica la directiva
# allow y depués deny.
<Directory /ftp/grupos/eso>
  <Limit STOR>
    order allow, deny
    AllowUser alumno
    Deny All
  </Limit>
</Directory>
# Necesario para NIS.
PersistentPasswd off
# Se establece como directorio raíz para los usuarios del grupo alumno,
# el directorio /ftp/grupos/eso.
DefaultRoot /ftp/grupos/eso alumno
#Se configura un servidor virtual por la IP 192.168.0.1
<VirtualHost 192.168.0.1>
  ServerName "Servidor Virtual. Acceso público a grupos ESO"
  DefaultRoot /ftp/grupos/eso
  Port 2121
  # Denegar el login (conexión) a todos los usuarios
  <Limit LOGIN>
    DenyAll
  </Limit>
  #Se configura el acceso anónimo para este servidor
  <Anonymous /ftp/grupos/eso>
    # En modo anónimo, el usuario asociado será ftp, y su grupo ftp
    User ftp
    Group ftp
    # No se obliga a que la cuenta ftp disponga de una shell válida
    RequireValidShell off
    # No se obliga a proporcionar password para el usuario anónimo
    AnonRequirePassword off
    # El usuario anónimo puede conectarse tanto como ftp como anonymous
    UserAlias anonymous ftp
    # Se permite el login a todos los usuarios (aplicándoles las
    # restricciones anteriores.
    <Limit LOGIN>
      AllowAll
    </Limit>
  </Anonymous>
</VirtualHost>
# Se incluyen las directivas para la configuración anónima del servidor principal
Include /etc/proftpd-anonymous.conf
```

Y el archivo /etc/proftpd-anonymous.conf es:

```
# Configuracion básica de acceso anónimo para el servidor ftp principal del centro.
<Anonymous /ftp/public>
  User ftp
  Group ftp
  UserAlias anonymous ftp
```

MaxClients 10
RequireValidShell off
AnonRequirePassword off
 # Se deniega escribir a cualquier usuario
<Limit WRITE>
DenyAll
</Limit>
</Anonymous>

Tenemos por tanto el siguiente esquema:

Directorio	Servidor	Modo de acceso	Descripción	Función
/home/<usuario>	Principal, puerto 21	login,password	No se permite acceso a los usuarios del grupo alumno y dptoinfo, al resto sí se le permite. Para conectarse deben dar su login y contraseña. Ningún usuario puede ir más arriba de su directorio home. No hay otras restricciones.	Asignar áreas privadas a los usuarios registrados del sistema.
/ftp/grupos/eso	Principal, puerto 21	login, password	Sólo se permite el acceso a los usuarios pertenecientes al grupo alumno. No pueden ir más arriba del directorio raíz. Tienen acceso a lectura y escritura.	Reserva un área exclusiva para un grupo de usuarios, en este caso, alumnos de ESO.
/ftp/public	Principal, puerto 21	anónimo	Permite acceso anónimo a un área de contenido público. Sólo se pueden descargar archivos, no está permitido la creación/borrado de directorios y archivos.	Este área servirá para contener datos de acceso público para su descarga (programas y manuales).
/ftp/grupos/eso	Virtual, puerto 2121	anónimo	Acceso al área del grupo de la ESO, pero para el resto de usuarios, en modo de acceso anónimo. Sólo se pueden descargar archivos, no está permitido la creación/borrado de directorios y archivos.	Este área ofrece la posibilidad de descargar/consultar la información almacenada en el área asignada al grupo de la ESO.

Reconsiderando el caso práctico

Llegados a este punto, podemos llegar a la conclusión de que quizás no sea demasiado recomendable estar haciendo uso de dos puertos para nuestro servidor FTP, y que deberíamos optar por un modelo de configuración más simple. Este modelo puede ser el que se plantea seguidamente:

- **/ftp/public** : Acceso anónimo. Sólo disponible para realizar descargas de programas (*/ftp/public/programas*) y manuales (*/ftp/public/manuales*).
- **/ftp/dpto/informatica** : Acceso restringido a usuarios pertenecientes al grupo *dptoinfo*. Posibilidad de realizar descargas y subir archivos, borrar/crear directorios y archivos. Debe quedar oculto al resto de usuarios del sistema y anónimos.
- **/ftp/grupos/eso** : Distinguiremos dos tipos de acceso, uno para usuarios pertenecientes al grupo *alumno*, el cual ofrecerá las mismas posibilidades que se ofrecen en */ftp/dpto/informatica* a los usuarios del grupo *dptoinfo*. Por otra parte, se dispondrá de un acceso para usuarios anónimos, permitiendo sólo descargas y cualquier otra operación que no suponga escribir o modificar nada del directorio.
- **/home/<usuario>** : Cualquier usuario que no pertenezca a los grupos *dptoinfo*, *alumno* o sea usuario anónimo, quedará ubicado al conectarse en su directorio *home*.

Para aplicar correctamente esta política de acceso será necesario haber creado correctamente las cuentas de usuario y grupos involucrados. Los usuarios a los que se les permitirá acceder privadamente a su directorio *home* no pueden pertenecer ni al grupo *dptoinfo* ni al grupo *alumno*. Entendemos por tanto que si hay una cuenta para cada alumno de la ESO, todos compartirán el directorio */ftp/grupos/eso* si pertenecen al mismo grupo. Esta idea puede ser buena al inicio del uso del servidor. Por otra parte si los profesores disponen de áreas privadas en el servidor, estas cuentas no podrán pertenecer al grupo *dptoinfo*, por lo que lo ideal es que todos usen una misma cuenta para acceder al área */ftp/dpto/informatica*, por ejemplo con una denominada *dptoinfo*, coincidiendo con el nombre del grupo al que pertenecerá. Esto se plantea de esta manera para facilitar el diseño posterior del archivo de configuración.

Los archivos de configuración quedarían del siguiente modo:

- **/etc/proftpd.conf**

```
# Número máximo de conexiones concurrentes
MaxInstances 30
# Usuario y grupo bajo el que se ejecutará el servidor.
User nobody
Group nogroup
# Todos los usuarios quedarán ubicados en su directorio home, salvo
# los pertenecientes al grupo alumno y dptoinfo.
DefaultRoot ~ !alumno !dptoinfo
# Se permite la sobreescritura de archivos en el directorio raíz de cada usuario.
<Directory />
  AllowOverwrite on
</Directory>
# Para el directorio /ftp/grupos/eso, se permite subir archivos a los
# usuarios del grupo alumno, y al resto no. Primero se aplica la directiva
# allow y depués deny.
<Directory /ftp/grupos/eso>
  <Limit STOR>
    order allow, deny
    AllowUser alumno
    Deny All
  </Limit>
</Directory>
```

```
# Para el directorio /ftp/dpto/informatica, se permite subir archivos a los
# usuarios del grupo dptoinfo, y al resto no. Primero se aplica la directiva
# allow y depués deny.
<Directory /ftp/dpto/informatica>
  <Limit STOR>
    order allow, deny
    AllowUser dptoinfo
    Deny All
  </Limit>
</Directory>
# Necesario para NIS.
PersistentPasswd off
# Se establece como directorio raíz para los usuarios del grupo alumno,
# el directorio /ftp/grupos/eso.
DefaultRoot /ftp/grupos/eso alumno
# Se establece como directorio raíz para los usuarios del grupo dptoinfo,
# el directorio /ftp/dpto/informatica.
DefaultRoot /ftp/dpto/informatica dptoinfo
# Se incluyen las directivas para la configuración anónima del servidor principal
Include /etc/proftpd-anonymous.conf
```

- **/etc/proftpd-anonymous.conf**

```
# A basic anonymous configuration, no upload directories.
<Anonymous /ftp>
  User ftp
  Group ftp
  # We want clients to be able to login with "anonymous" as well as "ftp"
  UserAlias anonymous ftp
  # Limit the maximum number of anonymous logins
  MaxClients 10
  # Don't make it require a valid password or shell.
  RequireValidShell off
  AnonRequirePassword off
  # We want 'welcome.msg' displayed at login, and '.message' displayed
  # in each newly chdired directory.
  DisplayLogin welcome.msg
  DisplayFirstChdir .message
  # Limit WRITE everywhere in the anonymous chroot
  <Limit WRITE>
    DenyAll
  </Limit>
  #La siguiente directiva posibilita que quede oculto el directorio
  # /ftp/dpto para todos los usuarios.
  <Directory /ftp/dpto>
    <Limit ALL>
      DenyAll
    </Limit>
  </Directory>
</Anonymous>
```

Parece que esta configuración es un poco menos compleja que la obtenida al principio.

Afinar los archivos de configuración

Podemos ahora ir afinando un poco estos archivos de configuración. Por ejemplo, con :

- **DisplayLogin** se muestra el contenido de un archivo de texto plano al usuario inmediatamente después de conectarse. El archivo puede referenciarse por el camino completo o bien de forma relativa. En este último caso, el archivo debe residir en el directorio raíz asignado al usuario. Si no se encuentra, no sucede nada, simplemente, no se le muestra nada al usuario. Podríamos crear un mensaje de bienvenida y almacenarlo en *welcome.msg*. Si este archivo es común para todos los posibles accesos del servidor, lo ubicaríamos en */ftp/welcome.msg*, por ejemplo.
- **DisplayQuit** se muestra el contenido de un archivo de texto plano al usuario inmediatamente antes de salir/desconectarse del servidor. El archivo puede referenciarse por el camino completo o bien de forma relativa. En este último caso se debe tener cuidado, pues se buscará el archivo en el directorio actual en el que se encontraba el usuario en el momento de solicitar la desconexión. Es por ello que se recomienda que el archivo esté referido de forma absoluta (por ejemplo, */ftp/bye.msg*).
- **AccessDenyMsg** se muestra un mensaje si el password es incorrecto. Cuando un usuario introduce un password incorrecto, el sistema le responde con un mensaje 530 de error indicando *Login incorrect*. Evidentemente, no es el login la causa del error, sino el password. Una manera de mejorar este mensaje de error es añadiendo la directiva *AccessDenyMsg*, acompañándola de un mensaje que haga alusión a que el error está en la contraseña suministrada.
- **AccessGrantMsg** obtenemos una función similar a *DisplayLogin*, salvo que en este caso lo que muestra es el texto que acompaña a la directiva. Puede emplearse la cadena especial *%u* para que, al mostrarse el mensaje, sea sustituída por el login del usuario.
- **MaxClientsPerHost** se controla el número máximo de conexiones por cada IP. Con esto podemos evitar que un usuario cree demasiadas conexiones y evite que el resto de usuarios puedan conectarse al servidor.

Más bloques de configuración

Limit

El protocolo FTP está basado en comandos que ejecuta el cliente de FTP sobre el servidor. Así, mediante estos comandos, el usuario conectado al srvidor podrá crear directorios, renombrarlos, subir archivos, descargarlos hacia su máquina local, borrar archivos, visualizar su contenido, etc. Cuando se emplea un cliente de FTP en modo gráfico, aunque resulte transparente para nosotros, realmente se está haciendo uso de estos comandos. Basta con echar un vistazo al área inferior de la ventana de gFTP para comprobar esto. Pues bien, cuando configuramos un servidor de FTP es normal determinar qué comandos pueden emplearse en determinados directorios, y cuáles no. Para ello se emplea la directiva *Limit*.

Esta directiva puede emplearse en diferentes contextos, es decir, a nivel de configuración principal del servidor, dentro del bloque de directivas asociadas a la configuración anónima del servidor, aplicada a un determinado directorio, o incluso a un servidor virtual.

Para ello se sigue la sintaxis *<Limit comando1 [comando2 ...]>*. Los comandos pueden ser cualesquiera de los utilizados en FTP, básicamente los siguientes:

- CWD : Se emplea para cambiar de directorio.
- MKD : Crea directorios.
- DELE : Borra un archivo.
- RMD : Borra un directorio.
- RETR : Descarga un archivo desde el servidor.

- STOR : Sube un archivo desde el cliente hacia el servidor.
- LOGIN : Permite aplicar restricciones sobre el proceso de login de un usuario. Sólo se aplica cuando se hace referencia a la configuración principal del servidor, a un servidor virtual (<*VirtualHost*>) o a la configuración de acceso anónimo (<*Anonymous*>). Este comando permitirá o no el login de ciertos usuarios.

Además se dispone de los siguientes grupos de comandos:

- READ : Engloba a todos los comandos relacionados con operaciones de lectura (RETR, SITE, SIZE, STAT).
- WRITE : Agrupa a todos los comandos relacionados con operaciones de escritura, creación o borrado (APPE, DELE, MKD, RMD, RNTO, STOR).
- DIRS : Reúne a todos los comandos que suponen procesos de listado de directorio (LIST, PWD, y otros).
- ALL : Todos los anteriores juntos.

A estos comandos se le pueden aplicar las siguientes directivas de restricción:

- **Allow [["from"] "all" | "none" | host | network [,host | network [,host | network ...]]]** y **Deny [["from"] "all" | "none" | host | network [,host | network [,host | network ...]]]** : Permite o no el uso de los comandos especificados por el bloque *Limit* a determinadas máquinas o redes (identificadas por su IP de red). Suele acompañarse de la directiva *order* que establece el orden de aplicación de *allow* y de *deny*. Un ejemplo:

```
<limit LOGIN>
  order allow, deny
  allow from 192.168.0
  deny from all
 </limit>
```

- **DenyAll** : Puede emplearse también dentro del bloque <Directory>y de <Anonymous>. Equivale a

```
order deny,allow
  deny from all
```

- **AllowAll** : Puede emplearse también dentro del bloque <Directory>y de <Anonymous>. Hace explícito el acceso completo al comando referido en <*Limit*>
- **DenyUser** : Deniega el acceso al comando a la lista de usuarios (separados por comas) que acompañe a esta directiva.
- **DenyGroup** : Deniega el acceso al comando a la lista de grupos (separados por comas) que acompañe a esta directiva.
- **AllowGroup** : Permite el acceso al comando a la lista de grupos (separados por comas) que acompañe a esta directiva.
- **AllowUser** : Permite el acceso al comando a la lista de usuarios (separados por comas) que acompañe a esta directiva.

Global

Todas las directivas que se ubiquen dentro de <*Global*> afectarán a todos los servidores configurados (tanto el principal como los virtuales). Pueden existir varios bloques <*Global*>, pero el servidor los considerará como uno solo. No todas las directivas pueden insertarse en este bloque.

Otros comandos de administración

ftpshut

Provoca la parada automática del servidor, cerrando las conexiones existentes e impidiendo nuevas conexiones. Este comando crea un archivo de control */etc/shutmsg*. Para que el servidor pueda de nuevo ponerse en marcha, será necesario ejecutar *ftpshut -R*, o simplemente borrar el archivo (que es lo que hace la opción -R). Puede establecerse un tiempo de espera antes de que se proceda a realizar la parada,

ftpshut +3 : El servidor se parará en 3 minutos.
ftpshut 1715 : El servidor se parará a las cinco y cuarto de la tarde.
ftpshut now : Se para inmediatamente.

ftptop

Muestra el estado de las conexiones FTP en tiempo real. Mientras se ejecuta, se deberá pulsar la tecla **q** para salir, y la **t** para cambiar entre los posibles modos de visualización de los datos.

ftpcount

Muestra el número de conexiones activas en un instante determinado.

ftpwho

Muestra información de los procesos que se están ejecutando asociados a las sesiones FTP abiertas.

Consideraciones finales

Conexión como root

ProFTPD no permite conexiones como *root* ya que esto puede originar una falla en la seguridad del sistema si alguien consigue la contraseña del root. Sin embargo, y si de manera excepcional se requiere conexión como root, deberá añadirse la directiva *RootLogin on* dentro del contexto que estimemos oportuno (configuración principal del servidor, servidor virtual, configuración del acceso anónimo o dentro del bloque <*Global*>.

Directiva TransferLog

Si deseamos emplear un archivo diferente al /var/log/xferlog para almacenar el registro de transferencias (descargas y subidas) efectuadas por los usuarios sobre el servidor, deberemos recurrir a la directiva *TransferLog* seguida de la ubicación del archivo de logs. Podemos ubicarla dentro del contexto que estimemos oportuno (configuración principal del servidor, servidor virtual, configuración del acceso anónimo o dentro del bloque <*Global*>.

Cómo expulsar a un usuario de nuestro servidor

A veces nos encontramos con un usuario que debe ser expulsado del servidor. Para ello procederemos de la siguiente manera:

1. Ejecutamos *ps aux | grep proftpd* para saber qué proceso tiene asignado, suponiendo que sabemos el nombre del usuario, claro.
2. *kill -9 <id-proceso-usuario>* donde <id-proceso-usuario>es el pid correspondiente al usuario que deseamos expulsar.

El archivo inicial /etc/proftpd.conf cuando se instala el servidor en Guadalinex.

```
  # /etc/proftpd.conf – This is a basic ProFTPD configuration file.
# To really apply changes reload proftpd after modifications.
#
ServerName "Debian"
ServerType standalone
DeferWelcome off
MultilineRFC2228 on
DefaultServer on
ShowSymlinks on
AllowOverwrite on
TimeoutNoTransfer 600
TimeoutStalled 600
TimeoutIdle 1200
DisplayLogin welcome.msg
DisplayFirstChdir .message
ListOptions "-l"
DenyFilter \*.*/
# Uncomment this if you are using NIS or LDAP to retrieve passwords:
#PersistentPasswd off
# Uncomment this if you would use TLS module:
#TLSEngine on
# Uncomment this if you would use quota module:
#Quotas on
# Uncomment this if you would use ratio module:
#Ratios on
# Port 21 is the standard FTP port.
Port 21
# To prevent DoS attacks, set the maximum number of child processes
# to 30. If you need to allow more than 30 concurrent connections
# at once, simply increase this value. Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances 30
# Set the user and group that the server normally runs at.
User nobody
Group nogroup
<Directory /*>
  # Umask 022 is a good standard umask to prevent new files and dirs
  # (second parm) from being group and world writable.
  Umask 022 022
  # Normally, we want files to be overwriteable.
  AllowOverwrite on
</Directory>
# A basic anonymous configuration, no upload directories.
<Anonymous ~ftp>
  User ftp
  Group nogroup
  # We want clients to be able to login with "anonymous" as well as "ftp"
  UserAlias anonymous ftp
  # Cosmetic changes, all files belongs to ftp user
  DirFakeUser on ftp
```

```
  DirFakeGroup on ftp
  RequireValidShell off
  # Limit the maximum number of anonymous logins
  MaxClients 10
  # We want 'welcome.msg' displayed at login, and '.message' displayed
  # in each newly chdired directory.
  DisplayLogin welcome.msg
  DisplayFirstChdir .message
  # Limit WRITE everywhere in the anonymous chroot
  <Directory *>
    <Limit WRITE>
      DenyAll
    </Limit>
  </Directory>
#   # Uncomment this if you're brave.
#   # <Directory incoming>
#   #   # Umask 022 is a good standard umask to prevent new files and dirs
#   #  # (second parm) from being group and world writable.
#      # Umask 022 022
#      # <Limit READ WRITE>
#        # DenyAll
#      # </Limit>
#      # <Limit STOR>
#       # AllowAll
#      # </Limit>
#   # </Directory>
#
# </Anonymous>
```

Capítulo 3

SERVIDOR DE CORREO. SENDMAIL. PARA QUÉ SIRVE. INSTALACIÓN. CONFIGURACIÓN.

3.1. Servidor de correo.

El servidor de correo es quizás uno de los servicios más utilizados de internet conjuntamente a los servidores de páginas web (apache), es más, lo que se tiende ultimamente es a unir ambas herramientas[1]en una que se denomina webmail. Pero vayamos por partes.

Antes de comenzar a instalar, y a configurar el primer servidor de correo vamos a repasar algunos conceptos previos.

Los componentes del sistema de correo son:

Mail User Agent (MUA) Un programa usado para crear y recibir mensajes de correo electrónico.

Mail Transfer Agent (MTA) El medio por el cual los mensajes de correo electrónico se transfieren de máquina en máquina hasta llegar a su destino.

Mail Delivery Agent (MDA) El programa que se encarga de depositar el mensaje de correo en el buzón del destinatario, una vez entregado por un MTA al servidor de correo.

Como ejemplos de programas dentro de cada una de ellas, podemos citar:

[MUA]Outlook, Eudora, Netscape Mail, Mozilla, Sylpheed, ximian-evolution, mail, elm, mutt, pine, mh/nmh.

[MTA]Sendmail, Postfix, smail, qmail, exim.

[MDA]mail, deliver, mail.local, procmail, fetchmail.

El mecanismo funciona de la siguiente manera. El Agente de Usuario confecciona el mensaje y lo envía mediante el protocolo SMTP a un Agente de Transporte. Este Agente de Transporte lo envía a través de la red a otros Agentes de Transporte, hasta que al final llega al Agente de Transporte que corresponde al destinatario del mensaje. El Agente de Entrega2.13 lo deposita en el buzón de correo electrónico del destinatario. Allí el mensaje espera hasta que el destinatario acceda con un Agente de Usuario mediante los protocolos POP o IMAP.

Vale y ¿cómo se encamina el correo? Es decir ¿como se deposita en un buzón determinado?

Cuando un Agente de Transporte recibe el encargo de transportar un mensaje de correo electrónico, supongamos que a la dirección linux@cepalcala.org, lo primero que hace es comprobar si es él mismo el encargado de manejar el correo para el dominio cepalcala.org. En caso de que no lo sea, le pregunta al sistema DNS qué máquina o máquinas son las encargadas de ello. Los registros MX2.14 son los que ofrecen esta información.

Por ejemplo, una consulta al DNS sobre los registros MX del dominio cica.es nos daría:

cica.es. 961 IN MX 15 mailgw2.cica.es.

[1] desde el punto de vista del usuario

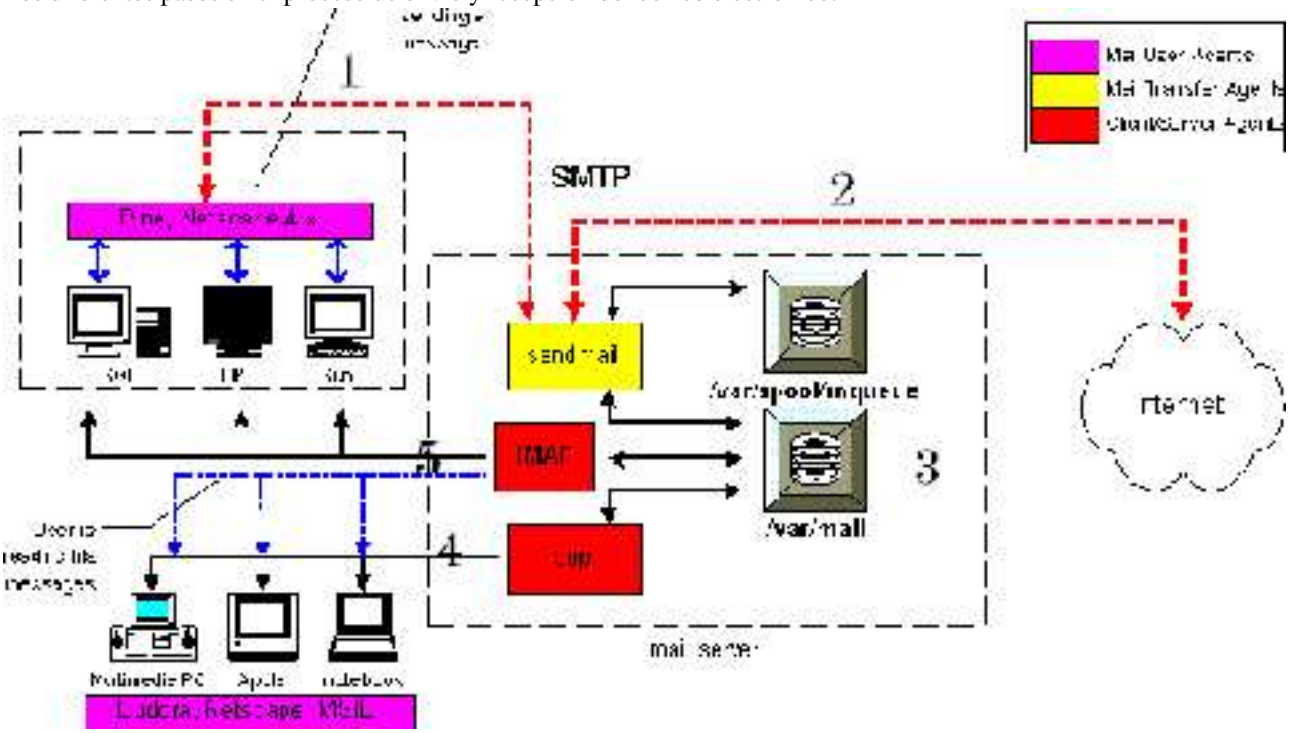
cica.es. 961 IN MX 20 mail.rediris.es.
cica.es. 961 IN MX 10 mailgw.cica.es.
Y sobre el dominio cepalcala.org sería:
cepalcala.org. 86400 IN MX 10 mx1.cepalcala.org.
cepalcala.org. 86400 IN MX 20 mx2.cepalcala.org.
Veamos qué implicaciones tienen estos datos en el correo.

Cada una de las líneas indica que el registro MX designa a una máquina que recibe correo para el dominio. De todas ellas, la preferente será la que tenga la prioridad más baja (el valor 10 será el preferido antes que el 15, y éste antes que el 20), y si no está disponible se irá al siguiente con menor prioridad. En este caso, si no hay ningún problema en la red o la máquina, mx1.cepalcala.org. será la máquina que recibirá los correos para la dirección linux@cepalacala.org.

En el caso de que no exista registro MX y el destino sea un host (por ejemplo cursolinux.cepalcala.org) también se le puede enviar correo electrónico a esa máquina concreta, como por ejemplo a la dirección linux@cursolinux.cepalcala.org

El correo puede pasar por varios Agentes de Transporte (MTA) hasta llegar a su destino.

Los diferentes pasos en el proceso de envío y recepción del correo electrónico.



En el paso 1, vemos que mediante un Agente de Usuario (pine, Netscape, Outlook...) componemos un mensaje y lo enviamos mediante el protocolo SMTP a un servidor de correo.

En el paso 2, el Agente de Transporte debe mirar a qué otro Agente de Transporte debe enviarlo en caso de que él no sea el receptor.

En el caso de que el correo para la dirección del destinatario lo gestione ese servidor de correo, en el paso 3 se guarda en el buzón correspondiente.

Cuando el destinatario quiere leer su correo, lo hace bien mediante el protocolo POP (paso 4) o el protocolo IMAP (paso 5).

¿Protocolo? Los protocolos se han comentado anteriormente y son los siguientes:

Protocolo SMTP

SMTP (Simple Mail Transfer Protocol) es un protocolo cliente-servidor basado en TCP. Su funcionamiento es muy simple. Una vez que se establece la conexión, el cliente envía comandos al servidor con la cabecera y el cuerpo del mensaje.

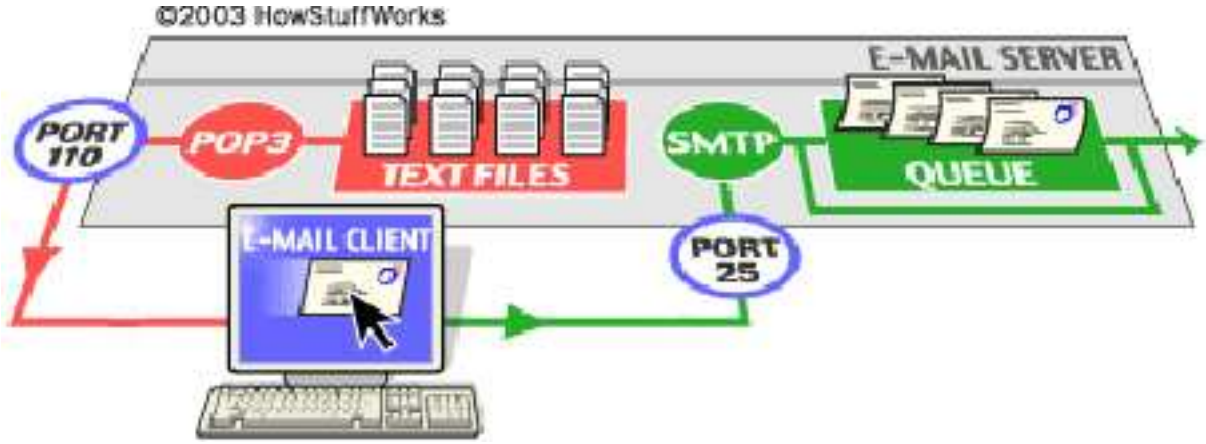
Este protocolo se basa en el envío de comandos de cuatro caracteres y códigos de respuesta de tres dígitos, más una serie de comentarios que lo hacen más legible. Acualmente se utiliza una versión conocida

como SMTP Extendido o ESMTP.

Protocolo POP

El protocolo POP (Post Office Protocol) se diseñó para permitir una gestión del correo sin tener que estar conectados continuamente con el servidor. La idea es conectarse con el servidor, descargarse al ordenador local los correos electrónicos y poder trabajar con ellos sin necesidad de estar conectados con el servidor continuamente, ni siquiera conectados a la red. Lo normal es que el correo al descargarlo, se borre del servidor, aunque hay opciones para conservarlo allí.

La siguiente figura es muy descriptiva de cómo el cliente de correo (MUA) envía al servidor (MTA) el correo al puerto 25 mediante el protocolo SMTP y lo recibe conectándose al puerto 110 mediante el protocolo POP.



Protocolo IMAP

El protocolo IMAP (Internet Messaging Access Protocol) es más potente que POP en la mayoría de los casos. En el modo desconectado (offline) sus capacidades son similares, pero es en el modo conectado (online) donde IMAP lo supera con creces. IMAP permite la manipulación de buzones en el servidor remoto como si fueran locales.

En conexiones de poco ancho de banda, permite capturar la estructura del mensaje sin descargarlo y seleccionar qué parte del mensaje nos interesa descargarnos.

Posee adicionalmente la capacidad de manipular un mensaje en el buzón remoto, permitiendo marcar los mensajes como leídos, borrados, contestados. La tendencia es a utilizar servidores con este protocolo en vez de POP. Pero claro está, esto depende de que nuestro proveedor del servicio de correo o administrador del sistema nos ofrezca esta posibilidad.

Si utilizamos un sistema de webmail y deseamos poder crear carpetas, éste es el protocolo adecuado.

3.1.1. Sendmail.

Sendmail ha sido una de las bestias negras de los administradores de sistemas Unix. Una relación amor-odio se entablaba con esta gran obra de ingeniería. Por una parte su potencia y capacidades eran insustituibles y por otra, su complejidad de configuración y sus errores de seguridad le hacían temible.

Por fortuna, esa situación ha cambiado. La inclusión del preprocesador de macros m4 para la configuración y sus reescrituras para mejorar el diseño y la seguridad, le hacen afrontar el futuro con optimismo. Es paradójico que uno de los ancianos más venerables de Internet, se mida a “jóvenes” como qmail o postfix y permanezca altivo y en forma.

Sendmail fue escrito por ERIC ALLMAN en la Universidad de California en Berkeley para el Unix de BSD. Ha sido portado a todas las plataformas existentes y todas las distribuciones de Linux la incorporan.

El estudio y configuración de sendmail es una de las cosas que distinguen a un auténtico "gurú" de UNIX. Realmente no es difícil, sino que la versatilidad y potencia de sendmail obliga al administrador a saber exactamente qué es lo que desea hacer. El estudio comparativo de diversas configuraciones es necesario hasta llegar aquella que cumpla plenamente con nuestras expectativas.

Hemos dejado a un lado el tema de seguridad. En tiempos Sendmail fué famoso por los múltiples agujeros que presentaba.

Por otro lado hay que destacar que sendmail depende íntimamente de la resolución de nombres DNS. De hecho existen unas entradas en las DNS denominadas MX (de Mail eXchanger) que son exhaustivamente consultadas por sendmail para decidir las rutas a seguir por los mensajes de correo. Una mala configuración del DNS puede hacer caer drásticamente el rendimiento de nuestro sistema de correo, o lo que es peor, hacer que este no funcione en absoluto.

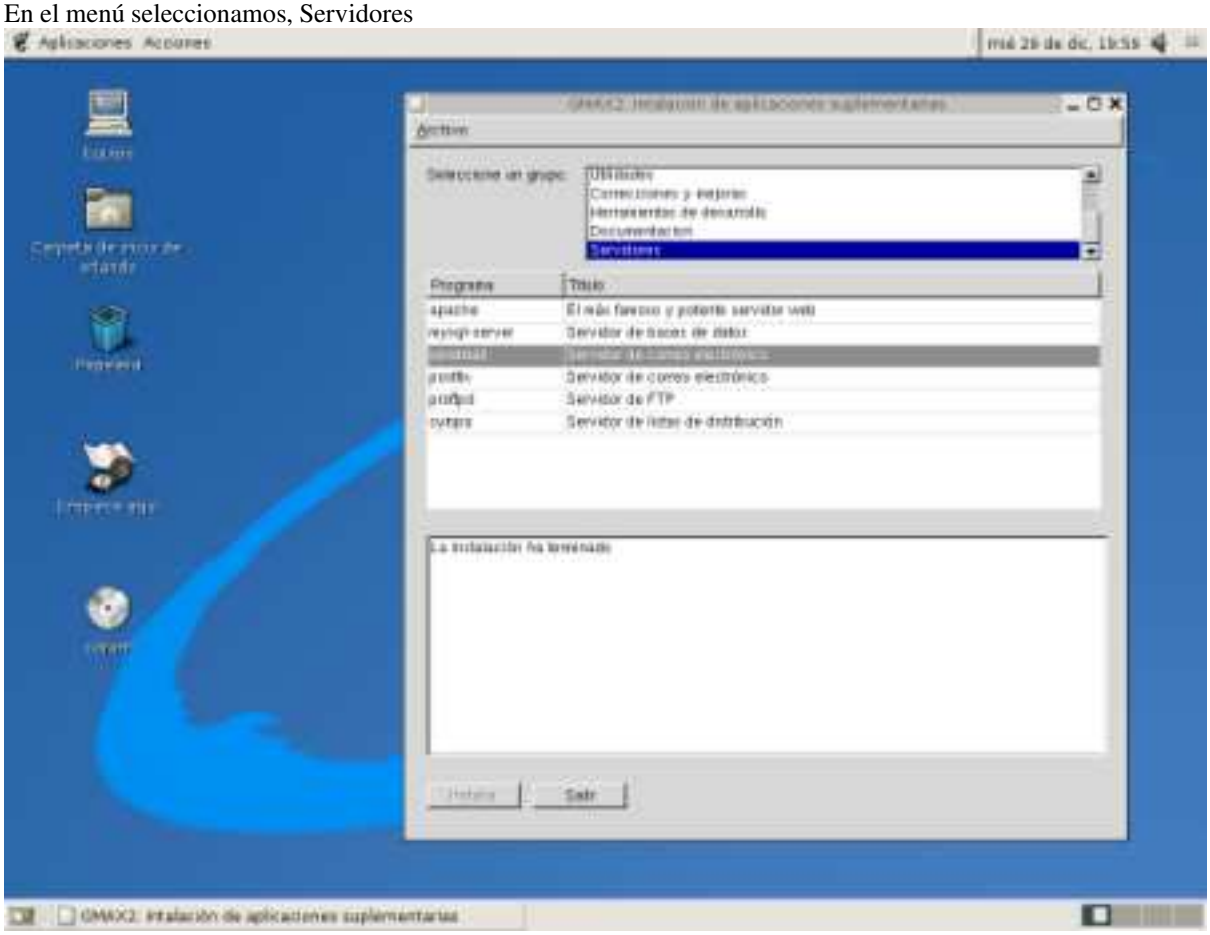
3.1.1.1. Instalación de Sendmail bajo guadalinex

En primer lugar vamos a realizar la instalación en formato gráfico, para ello debemos de disponer del Cd suplementario de Guadalinex, que se puede descargar desde

ftp://ftp.cica.es/mirrors/Linux/Guadalinex/descargas/iso/suplementos-g2004-vol1.iso

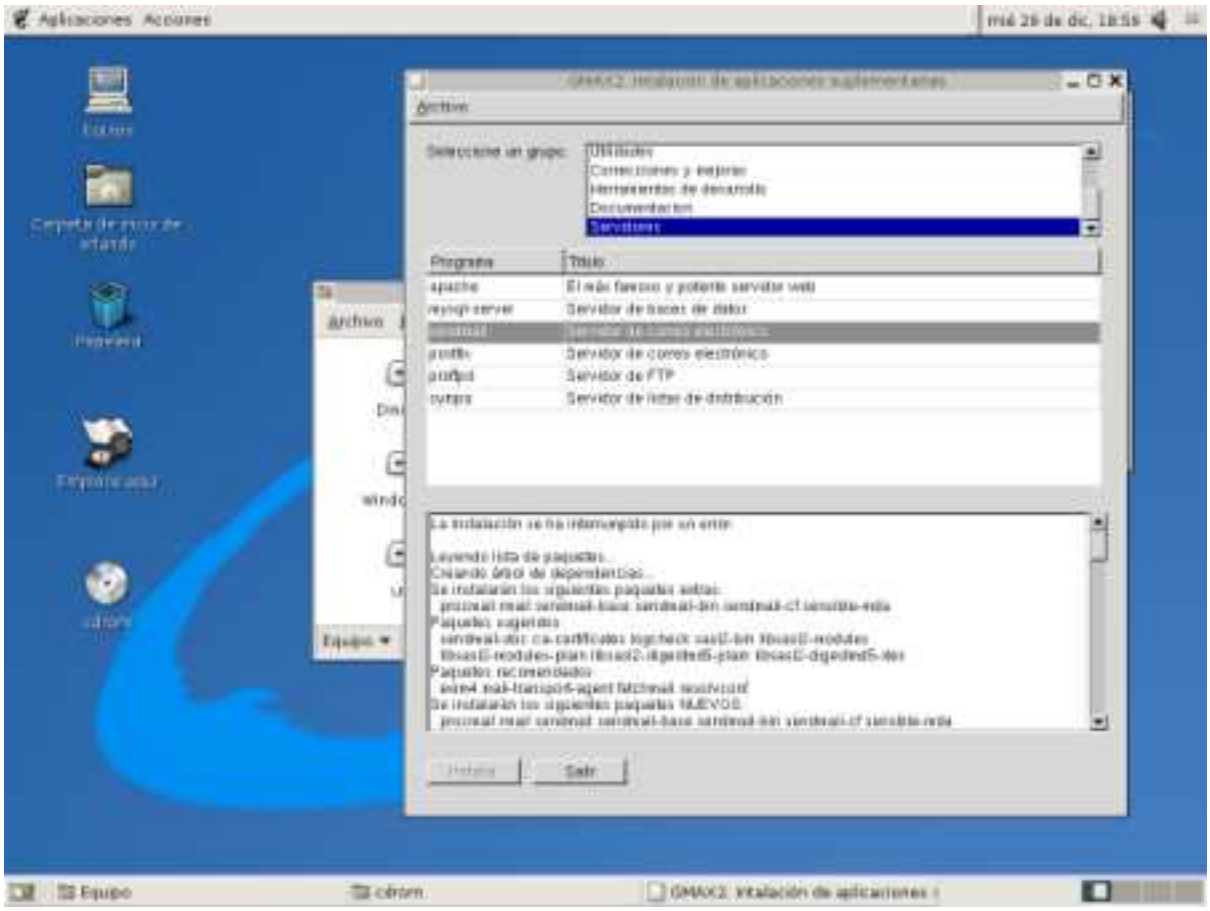
o en cualquiera de sus mirrors.

En el menú seleccionamos, Servidores



Durante el proceso se instalarán ademas librerias adicionales, las cuales bastará con dejar en las preguntas que se nos realice las aquellas opciones que se encuentren por defecto.

Finalmente deberemos de obtener una pantalla en la que se nos indica que todo ha ido correctamente:



A continuación realizaremos la instalación desde el modo consola. Para instalar los paquetes que necesitamos para el correo utilizaremos nuestro aliado: apt-get.

Si vamos a utilizar sendmail, es importante utilizar una versión reciente, ya que son muchas las mejoras de seguridad que incorporan. Por ejemplo, el que vamos a instalar de prueba, será sendmail-8.13.1-13 Con la utilidad apt-get, obtendremos una versión actualizada.

Los paquetes que instalaremos serán: sendmail, el agente de transporte, sendmail-cf, las utilidades para la configuración e imap, que es el paquete en que se encuentran los servidores de POP e IMAP.

root@guadalinex:/home/orlando# **apt-get install sendmail uw-imapd ipopd**
Leyendo lista de paquetes... Hecho
Creando Ã¡rbol de dependencias... Hecho
Se instalarÃ¡n los siguientes paquetes extras:
libc-client2002edebian mlock rmail sendmail-base sendmail-bin sendmail-cf
sensible-mda
Paquetes sugeridos:
logcheck uw-mailutils sendmail-doc ca-certificates sasl2-bin
libsasl2-modules libsasl2-modules-plain libsasl2-digestmd5-plain
libsasl2-digestmd5-des mutt imap-client
Paquetes recomendados
exim4 mail-transport-agent resolvconf
Se instalarÃ¡n los siguientes paquetes NUEVOS:
ipopd libc-client2002edebian mlock rmail sendmail sendmail-base sendmail-bin
sendmail-cf sensible-mda uw-imapd
0 actualizados, 10 se instalarÃ¡n, 0 para eliminar y 116 no actualizados.
Necesito descargar 2690kB de archivos.
Se utilizarÃ¡n 6640kB de espacio de disco adicional despuÃ©s de desempaquetar.

Â¿Desea continuar? [S/n] s
Des:1 http://http.guadalinex.org sarge/main mlock 7:2002edebian1-3 [19,4kB]
Des:2 http://http.guadalinex.org sarge/main libc-client2002edebian 7:2002edebian1-3 [574kB]
Des:3 http://http.guadalinex.org sarge/main ipopd 7:2002edebian1-3 [34,3kB]
Des:4 http://http.guadalinex.org sarge/main uw-imapd 7:2002edebian1-3 [61,2kB]
Des:5 http://http.guadalinex.org sarge/main sendmail-bin 8.13.1-13 [806kB]
Des:6 http://http.guadalinex.org sarge/main rmail 8.13.1-13 [216kB]
Des:7 http://http.guadalinex.org sarge/main sendmail-base 8.13.1-13 [331kB]
Des:8 http://http.guadalinex.org sarge/main sendmail-cf 8.13.1-13 [271kB]
Des:9 http://http.guadalinex.org sarge/main sensible-mda 8.13.1-13 [191kB]
Des:10 http://http.guadalinex.org sarge/main sendmail 8.13.1-13 [186kB]
Descargados 2690kB en 56s (47,6kB/s)
Preconfiguring packages ...
Seleccionando el paquete mlock previamente no seleccionado.
(Leyendo la base de datos ...
98647 ficheros y directorios instalados actualmente.)
Desempaquetando mlock (de .../mlock_7 %3a2002edebian1-3_i386.deb) ...
Seleccionando el paquete libc-client2002edebian previamente no seleccionado
Una vez instalados, tecleando
root@guadalinex:/# **update-rc.d sendmail defaults**

Podremos configurarlos para que arranquen automáticamente, si bien en Debian (Guadalinex) esto debería realizarse de forma automática. Si aparece el siguiente mensaje “System startup links for /etc/init.d/sendmail already exist.” es que los servicios ya están activos en el arranque.

Los servicios que activaremos serán:

sendmail Demonio del Agente de Transporte. Utiliza el puerto 25.

imap Servidor para acceder a los buzones de usuario utilizando el protocolo IMAP. Utiliza el puerto 143.

imaps Igual que imap pero con un protocolo cifrado. Utiliza el puerto 993.

ipop3 Servidor del protocolo POP. Utiliza el puerto 110.

ipop3s Servidor POP seguro. Utiliza el puerto 995.

3.1.1.2. Configuración de Sendmail

Pasemos a configurar sendmail. Esta configuración será independiente del tipo de instalación que se haya realizado anteriormente. Miremos el fichero sendmail.cf, que se encuentra en el directorio /etc/mail.

```
# strip group: syntax (not inside angle brackets!) and trailing semicolon
R$* $: $1 <@>mark addresses
R$* <$* >$* <@>$: $1 <$2 >$3 unmark <addr>
R@ $* <@>$: @ $1 unmark @host:...
R$* [ IPv6 : $+ ] <@>$: $1 [ IPv6 : $2 ] unmark IPv6 addr
R$* :: $* <@>$: $1 :: $2 unmark node::addr
R:include: $* <@>$: :include: $1 unmark :include:...
R$* : $* [ $* ] $: $1 : $2 [ $3 ] <@>remark if leading colon
R$* : $* <@>$: $2 strip colon if marked
R$* <@>$: $1 unmark
R$* ; $1 strip trailing semi
R$* <$+ :; >$* $@ $2 :; <@>catch <list:;>
R$* <$* ; >$1 <$2 >bogus bracketed semi
```

Como véis, quien sea capaz de entender esto, no debe ser una persona normal. Es una de las razones de la mala fama (y hasta merecida) de sendmail.

Pero gracias a la utilización del preprocesador de macros m4, la tarea se nos ha vuelto más fácil. Bueno, aún así nos llevará un poco comprenderla totalmente. Nuestro fichero de configuración será /etc/mail/sendmail.mc y a partir de él obtendremos el fichero sendmail.cf, que es el que leerá sendmail.

En el directorio /usr/share/sendmail/cf/debian existe el fichero .mc para Debian.

Tenemos un punto a nuestro favor. La configuración por defecto nos servirá casi sin modificaciones en un tanto por ciento muy elevado de casos. Cuando lo tengamos a nuestro gusto, simplemente tecleamos #make en el directorio /etc/mail y se generará automáticamente el fichero sendmail.cf, en cuyo caso debermos obtener una pantalla similar a esta:

root@guadalinex:/etc/mail# make
Updating auth ...sasl2-bin not installed, not configuring sendmail support.
To enable sendmail SASL2 support at a later date, invoke"/usr/share/sendmail/update_auth"
Updating databases ...
Reading configuration from /etc/mail/sendmail.conf.
Validating configuration.
Creating /etc/mail/databases...
Creating /etc/mail/relay-domains
#Optional file...
Updating Makefile ...
Validating configuration.
Creating /etc/mail/Makefile...
Updating sendmail.cf ...
The following file(s) have changed:
/etc/mail/sendmail.cf
** ** You should issue '/etc/init.d/sendmail reload' ** **

Como bien nos indica la última linea de los mensajes anteriores para que tome la nueva configuracion deberemos de teclear (EJECUTAR)

#/etc/init.d/sendmail reload

Lo cual realizará la carga nuevamente del agente de correo.

¿Que debemos modifcar en el fichero anterior? Comentaremos las líneas más interesantes del mismo.[2]

Para trabajar con el fichero se puede hacerse con el vi o con cualquier otro editor de textos, eso si como root.

Todo fichero de definición tendrá los siguientes campos:

Una entrada OSTYPE, que defina el sistema operativo. El valor "unknown" deja al make la opción de escogerlo automáticamente

Una o varias FEATURE's que definen las funcionalidades que va a tener esta configuración

Diversas entradas de declaración de los mailers que va a tener definidos este sistema. Al menos un mailer debe estar definido Declaracion opcional de los diversos "trucos" que utiliza el sistema. En el ejemplo de RedHat, los hacks definen las diversas opciones que controlan el acceso y verificación de los remitentes y destinatarios del correo

En el caso de querer especificar una línea del sendmail,cf de manera literal, declararemos la entrada LOCAL_RULESET_XXXX y a continuación aquello que queramos que figure en el fichero

Por motivos de espacio es imposible detallar todas las opciones, features, y hacks pre-definidos. Existe un fichero /usr/lib/sendmail.cf/README que incluye la documentación completa, mas otro /usr/lib/sendmail-cf/README.check que define los diversos hacks relacionados con el control del enrutamiento del correo y de las reglas de aceptación de este, para evitar el denominado "mail spamming" o abuso de nuestro servidor de correo para el envio indiscriminado de mail. Como siempre, la recomendación es probar y experimentar antes de dar una configuración por válida

De hecho este fichero que incorpora sendmail aunque funcionará con casí toda seguridad en un 99 % de los casos sin modifcación alguna, es uno de los ficheros mas pequeños con los cuales podremos encontrarnos por lo que además de las líneas que este contiene se comentarán algunas que si se desean añadir al fichero puede hacerse y se verán para que sirve.[3]

[root@linux mail]# vi sendmail.mc
divert(-1)dnl

[2] Recordamos que vamos a mostrar el fichero /etc/mail/sendmail.mc, pero ya sabe alguien si prefiere directamente trabajar con el otro ;)

[3] Si estas leyendo este manual en papel y en blanco y negro posiblemente te costará diferenciar las explicaciones ya que se han añadido en color rojo entre las líneas del fichero.

```
#------------------------------------------------------------
# $Sendmail: debproto.mc,v 8.13.1 2004-09-12 18:29:33 cowboy Exp $
#
# Copyright (c) 1998-2004 Richard Nelson. All Rights Reserved.
#
# cf/debian/sendmail.mc. Generated from sendmail.mc.in by configure.
#
# sendmail.mc prototype config file for building Sendmail 8.13.1
#
# Note: the .in file supports 8.7.6 - 9.0.0, but the generated
#file is customized to the version noted above.
#
# This file is used to configure Sendmail for use with Debian systems.
#
# If you modify this file, you will have to regenerate /etc/mail/sendmail.cf
# by running this file through the m4 preprocessor via one of the following:
#* ‘sendmailconfig‘
#* ‘make‘
#* ‘m4 /etc/mail/sendmail.mc >/etc/mail/sendmail.cf‘
# The first two options are preferred as they will also update other files
# that depend upon the contents of this file.
#
# The best documentation for this .mc file is:
# /usr/share/doc/sendmail-doc/cf.README.gz
#
#------------------------------------------------------------
divert(0)dnl
#
# Copyright (c) 1998-2004 Richard Nelson. All Rights Reserved.
#
# This file is used to configure Sendmail for use with Debian systems.
#
```

Las líneas divert y dnl son comentarios.

```
define(‘_USE_ETC_MAIL_’)dnl
include(‘/usr/share/sendmail/cf/m4/cf.m4’)dnl
```

Carga el fichero cf.m4 que necesita.

```
VERSIONID(‘$Id: sendmail.mc, v 8.13.1-13 2004-09-12 18:29:33 cowboy Exp $’)
OSTYPE(‘debian’)dnl
DOMAIN(‘debian-mta’)dnl
```

Decimos la versión y el sistema operativo. Le servirá para adoptar opciones personalizadas. En este caso, cargará el fichero /usr/share/sendmail/cf/debian/debian-mta.m4.

```
dnl # Items controlled by /etc/mail/sendmail.conf - DO NOT TOUCH HERE
undefine(‘confHOST_STATUS_DIRECTORY’)dnl #DAEMON_HOSTSTATS=
dnl # Items controlled by /etc/mail/sendmail.conf - DO NOT TOUCH HERE
dnl #
dnl # General defines
dnl #
dnl # SAFE_FILE_ENV: [undefined] If set, sendmail will do a chroot()
dnl #into this directory before writing files.
dnl #If *all* your user accounts are under /home then use that
dnl #instead - it will prevent any writes outside of /home !
dnl # define(‘confSAFE_FILE_ENV’, ‘’)dnl
dnl #
dnl #
```

dnl # Uncomment and edit the following line if your outgoing mail needs to
dnl # be sent out through an external mail server:
dnl #
dnl define(‘SMART_HOST’,‘smtp.miproveedor.com’)

La opción que aparece en azul no aparece en el fichero original de Guadalinex y que se añade al fichero como comentada sirve para indicar que le paso la tarea del correo a smtp.miproveedor.com, es decir es un agente de transporte al que le paso la pelota y que él se encargue de los siguientes pasos. Muy útil si estamos en una red privada y solamente ese “host inteligente” puede salir hacia el exterior. Para utilizarla, tendríamos que poner cuál es ese host en nuestro caso y descomentarla quitando el dnl.

dnl # Daemon options - restrict to servicing LOCALHOST ONLY !!!
FEATURE(‘no_default_msa’)dnl
dnl DAEMON_OPTIONS(‘Family=inet6, Name=MTA-v6, Port=smtp, Addr=::1’)dnl
DAEMON_OPTIONS(‘Family=inet, Name=MTA-v4, Port=smtp, Addr=127.0.0.1’)dnl
dnl DAEMON_OPTIONS(‘Family=inet6, Name=MSP-v6, Port=submission, Addr=::1’)dnl
DAEMON_OPTIONS(‘Family=inet, Name=MSP-v4, Port=submission, Addr=127.0.0.1’)dnl

Aquí le dices que funcione solo para la maquina local, no recibira ningún correo para enviar cuyo origen difiera de 127.0.0.1 Muy importante. Debemos poner nuestra dirección de red local en vez de 127.0.0.1 si queremos que los clientes puedan comunicar con el servidor.

dnl # Be somewhat anal in what we allow
define(‘confPRIVACY_FLAGS’,dnl
‘needmailhelo,needexpnhelo,needvrfyhelo,restrictqrun,restrictexpand,nobodyreturn,authwarnings’)dnl
dnl #
dnl # Define connection throttling and window length
define(‘confCONNECTION_RATE_THROTTLE’, ‘15’)dnl
define(‘confCONNECTION_RATE_WINDOW_SIZE’,‘10m’)dnl
dnl #
dnl # Features
dnl #
dnl # The access db is the basis for most of sendmail’s checking
FEATURE(‘access_db’, , ‘skip’)dnl
dnl #
dnl # The greet_pause feature stops some automail bots - but check the
dnl # provided access db for details on excluding localhosts...
FEATURE(‘greet_pause’, ‘1000’)dnl 1 seconds
dnl #
dnl # Delay_checks allows sender<->recipient checking
FEATURE(‘delay_checks’, ‘friend’, ‘n’)dnl
dnl #
dnl # If we get too many bad recipients, slow things down...
define(‘confBAD_RCPT_THROTTLE’,‘3’)dnl
dnl #
dnl # Stop connections that overflow our concurrent and time connection rates
FEATURE(‘conncontrol’, ‘nodelay’, ‘terminate’)dnl
FEATURE(‘ratecontrol’, ‘nodelay’, ‘terminate’)dnl
dnl # Masquerading options
FEATURE(‘always_add_domain’)dnl
MASQUERADE_AS(‘sid’)dnl
FEATURE(‘allmasquerade’)dnl
FEATURE(‘masquerade_envelope’)dnl
dnl #

Siempre añade el dominio para completar las direcciones de correo electrónico. Por ejemplo, estando en el dominio midominio.org, un correo enviado al usuario linux, se completará como linux@midominio.org.

Aquí termina el fichero de configuración de guadalinex original. Comentemos algunas cosas más.

dnl # Desde aqui no debe venir en el fichero original de guadalinex pero si otros ficheros o incluso
dnl # añadirlo nosotros en el nuestro
define(‘confDEF_USER_ID’,“8:12”)dnl
Usuario y grupo que ejecutarán el proceso sendmail (normalmente usuario mail y grupo mail).
define(‘confTO_CONNECT’, ‘1m’)dnl
define(‘confTRY_NULL_MX_LIST’,true)dnl
define(‘confDONT_PROBE_INTERFACES’,true)dnl
define(‘PROCMAIL_MAILER_PATH’,‘/usr/bin/procmail’)dnl
define(‘ALIAS_FILE’, ‘/etc/aliases’)dnl
Cuál es el fichero de alias. Crea una dirección de correo virtual y la asocia a otra dirección. Por ejemplo la línea webmaster: admin dice que todos los correos que vayan a la dirección webmaster@dominio-configurado.org, siendo dominio-configurado.org el que está recogiendo nuestro sendmail, vayan a la dirección admin@dominio-configurado.org
dnl define(‘STATUS_FILE’, ‘/etc/mail/statistics’)dnl
define(‘UUCP_MAILER_MAX’, ‘2000000’)dnl
define(‘confUSERDB_SPEC’, ‘/etc/mail/userdb.db’)dnl
define(‘confPRIVACY_FLAGS’, ‘authwarnings,novrfy,noexpn,restrictqrun’)dnl
define(‘confAUTH_OPTIONS’, ‘A’)dnl
dnl #
dnl # The following allows relaying if the user authenticates, and disallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(‘confAUTH_OPTIONS’, ‘A p’)dnl
dnl #
dnl # PLAIN is the preferred plaintext authentication method and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl #
dnl TRUST_AUTH_MECH(‘EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN’)dnl
dnl define(‘confAUTH_MECHANISMS’, ‘EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN’)dnl
dnl # Rudimentary information on creating certificates for sendmail TLS:
dnl # make -C /usr/share/ssl/certs usage
dnl #
dnl define(‘confCACERT_PATH’,‘/usr/share/ssl/certs’)
dnl define(‘confCACERT’,‘/usr/share/ssl/certs/ca-bundle.crt’)
dnl define(‘confSERVER_CERT’,‘/usr/share/ssl/certs/sendmail.pem’)
dnl define(‘confSERVER_KEY’,‘/usr/share/ssl/certs/sendmail.pem’)
dnl #
dnl # This allows sendmail to use a keyfile that is shared with OpenLDAP’s
dnl # slapd, which requires the file to be readble by group ldap
dnl #
dnl define(‘confDONT_BLAME_SENDMAIL’,‘groupreadablekeyfile’)dnl
dnl #
dnl define(‘confTO_QUEUEWARN’, ‘4h’)dnl
Los mensajes que recoge sendmail los pone en una cola (un directorio en donde los va guardando) y los envía cuando puede. Por ejemplo, si nuestra conexión a Internet no es permanente o el agente de transporte destino no está operativo. Este parámetro designa el tiempo (4 horas) que al cumplirse, nos envía un mensaje indicando que no lo ha podido entregar.
dnl define(‘confTO_QUEUERETURN’, ‘5d’)dnl
Si en 5 días no ha conseguido entregarlo al destinatario, nos lo devuelve.
dnl define(‘confQUEUE_LA’, ‘12’)dnl
dnl define(‘confREFUSE_LA’, ‘18’)dnl

define(‘confTO_IDENT’, ‘0’)dnl
dnl FEATURE(delay_checks)dnl
FEATURE(‘no_default_msa’,‘dnl’)dnl
FEATURE(‘smrsh’,‘/usr/sbin/smrsh’)dnl
FEATURE(‘mailertable’,‘hash -o /etc/mail/mailertable.db’)dnl
FEATURE(‘virtusertable’,‘hash -o /etc/mail/virtusertable.db’)dnl
FEATURE(redirect)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
dnl # The -t option will retry delivery if e.g. the user runs over his quota.
dnl #
FEATURE(local_procmail,‘’,‘procmail -t -Y -a $h -d $u’)dnl
FEATURE(‘access_db’,‘hash -T<TMPF>-o /etc/mail/access.db’)dnl
FEATURE(‘blacklist_recipients’)dnl
EXPOSED_USER(‘root’)dnl
FEATURE(‘accept_unresolvable_domains’)dnl
dnl FEATURE(‘relay_based_on_MX’)dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
LOCAL_DOMAIN(‘localhost.localdomain’)dnl
dnl #
dnl # The following example makes mail from this host and any additional
dnl # specified domains appear to be sent from mydomain.com
dnl #
dnl MASQUERADE_AS(‘mydomain.com’)dnl

Si nuestra máquina se llama mail.midominio.org, el correo que salga de ella, si no hacemos algo en contrario, será con direcciones del tipo: usuario@mail.midominio.org. Si queremos que salgan con direcciones del dominio, es decir, usuario@midominio.org, utilizamos el enmascaramiento. Es normal dentro de una organización utilizar el dominio para el correo, y no direcciones de máquinas particulares.

dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
dnl FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomainalias.com as well
dnl #
dnl FEATURE(masquerade_entire_domain)dnl
dnl #
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
MAILER(smtp)dnl
MAILER(procmail)dnl

Hasta aquí la configuración de sendmail. Lo conveniente es no tocar el fichero sendmail.cf, sino todos los cambios realizarlos sobre el fichero sendmail.mc y de este generar el fichero sendmail.cf.

Abrimos ahora el archivo /etc/mail/access y agregamos algunas líneas para definir quienes podrán hacer uso de nuestro servidor de correo para poder enviar mensajes:

por defecto solo se permite enviar correo
a localhost...
localhost.localdomain RELAY
localhost RELAY

127.0.0.1 RELAY
miredlocal.org.mx RELAY
linux.miredlocal.org.mx RELAY
etc.
#
Y también podemos agregar las direcciones de correo
electrónico de aquellos a quienes consideremos
"indeseables", o que queramos bloquear.
Spam@algun_Spamer.com REJECT
info@otro_Spammer.com REJECT
etc.

3.1.1.3. Instalación en Mandrake.

Como ya a estas alturas conocereís el Panel de Control la instalación gráfica no la realizaremos y nos basaremos en la instalación en modo consola, si alguno lo prefiere siempre puede usar el Centro de Control de Mandrake.

En principio para la instalación se puede realizar con el famoso rpm que ya se ha utilizado en otras ocasiones pero como desconozco el número de paquetes que cada uno tiene en su ordenador vamos a utlizar otra herramienta similar que nos ayudará a instalar los paquetes dependendientes: urmpi.

¿Como se utiliza urmpi? La herramienta urpmi es el elemento central de todo el conjunto urpmi. Esta herramienta permite la instalación de rpms conocidos y resuelve las dependecias basándose en las bases de datos de los paquetes. La invocación simple de este comando es para instalar un paquete, esto se hace así:

urpmi vim - esto instalará el paquete vim y todos aquellos paquetes de cuales dependa.

urpmi también intenta ayudar si no conoce el nombre exacto del paquete. Por ejemplo, si quiere instalar el sistema de preparación de documentos DocBook y no conoce que paquetes necesita, puede hacerse lo siguiente;

[root@deneb root]#urpmi docbook

The following packages contain docbook: docbook-dtd31-sgml docbook-dtd412-xml koffice docbook-style-dsssl

docbook-style-dsssl-doc docbook-style-xsl docbook-utils docbook-dtd41-sgml

[root@deneb root]#urpmi docbook-dtd41-sgml

To satisfy dependencies, the following packages are going to be installed (1 MB):

libxml2-utils-2.4.16-2mdk.i586 docbook-dtd41-sgml-1.0-5mdk.noarch sgml-common-0.6.3-4mdk.noarch

Is it OK? (Y/n)

[... listing trimmed ...]

Pero urmpi no esta solo, contiene además otro conjunto de “herramientas” que nos ayudarán, estas son: urpmq, urpmf, urpmi.addmedia,...

urpme

El comando urpme es similar al comando urpmi, excepto que elimina los paquetes instalados. Éste también le preguntará eliminar todos aquellos paquetes que sean dependientes del paquete que se va a eliminar. Por ejemplo, para desintalar samba-common con el comando urpme samba-common da lo siguiente:

[root@deneb root]#urpme samba-common

To satisfy dependencies, the following packages are going to be removed (14 MB):

samba-common-2.2.3a-10mdk samba-2.2.3a-10mdk samba-client-2.2.3a-10mdk

Is it OK? (Y/n)

[... listing trimmed ...]

urpmq

El comando urpmq le permite buscar paquetes. Se le proporciona un término de búsqueda y urpmq intentará encontrar el nombre de los paquetes que contengan ese término. Se devolverán resultados de lista de páquetes sean tanto paquetes instalados como no. Así, por ejemplo, si quiere conocer que paquetes tienen relación con el kernel, puede utilizar el comando urpmq kernel, esto hará lo siguiente

[root@deneb root]#urpmq kernel

The following packages contain kernel: kernel-source kernel-headers kernel22

kernel-secure-2.4.18.6mdk kernel-doc-pdf kernel-doc-ps kernel-doc kernel-alert
kernel22-smp fortune-kernelcookies kernel-2.4.18.6mdk kernel-doc-html
kernel-enterprise-2.4.18.6mdk kernel-smp-2.4.18.6mdk

urpmf

El comando urpmf es una herramiente de búsqueda más avanzada que le permitirá buscar un archivo en todos los paquetes conocidos (aquellos instalados como disponibles). Así, por ejemplo, si prueba de compilar un programa y el script de configuración se queja de no encontrar ncurses.h, puede hacer urpmf ncurses.h para encontrar que éste es parte del paquete libncurses5-devel (entonces después ústed puede escribir urpmi libncurses5-devel para instalarlo si lo quiere).

[root@deneb root]#urpmf ncurses.h
libncurses5-devel:/usr/include/ncurses.h
libncurses5-devel:/usr/include/ncurses/ncurses.h
php-devel:/usr/src/php-devel/extensions/ncurses/php_ncurses.h

urpmi.addmedia

Este comando le permite añadir nuevas fuentes de rpms o sus bases de datos urpmi. Si la fuente és un recurso remote, éste ha de tener los archivos hdblist adecuados (los mirrors de Mandrake tienen estos archivos, otros puede que no). Ústed puede usar este comando para añadir fuentes que están ubicadas en un CD o disco duro. Una lista de fuentes oficiales de Mandrake está disponible en: Mandrake web site o bien utilizar alguna de estas que os apunto para mandrake 9.1:

urpmi.addmedia –update updates ftp://ftp.rediris.es/pub/linux/distributions/mandrakelinux/old/updates/9.1/RPMS with ../base/hdlist.cz

urpmi.addmedia plf ftp://ftp.cica.es/mirrors/Linux/plf/mandrake/9.1 with hdlist.cz

urpmi.addmedia main ftp://ftp.rediris.es/pub/linux/distributions/mandrakelinux/official/9.1/i586/Mandrake/RPMS with ../base/hdlist.cz

urpmi.addmedia contrib ftp://ftp.cica.es/pub/Linux/Mandrakelinux/official/9.1/contrib/i586 with ../../i586/Mandrake/base/hd

urpmi.addmedia plf ftp://knight.zarb.org/pub/plf/9.1 with hdlist.cz

urpmi.addmedia main ftp://public.ftp.planetmirror.com/pub/mandrake/9.1/i586/Mandrake/RPMS with ../base/hdlist.cz

urpmi.addmedia contrib ftp://ftp.tugraz.at/mirror/Mandrake-linux/Mandrake/9.1/contrib/RPMS with ../../i586/Mandrake/bas

urpmi.addmedia jpackage.free ftp://ftp.pbone.net/pub/jpackage/1.5/mandrake-9.1/free with hdlist.cz

urpmi.addmedia –update updates http://distro.ibiblio.org/pub/linux/distributions/mandrake/Mandrake/updates/9.1/RPMS/ with ../base/hdlist.cz

urpmi.addmedia texstar ftp://ftp.ibiblio.org/pub/Linux/distributions/contrib/texstar/mandrake/9.1/rpms with hdlist.cz

Por cierto para aquellos que están utilizando la versión 10 ó 10.1 se pueden dirigir a la siguiente web:

http://easyurpmi.zarb.org/

en la que encontrarán las direcciones urpmi correspondientes a su distribucion, entre las cuales la de cica es la siguiente para Mandrake 10.1:

urpmi.addmedia contrib ftp://ftp.cica.es/pub/Linux/Mandrakelinux/official/10.1/i586/media/contrib with media_info/hdlist.cz

Como ejemplo se puede añadir una fuente que contiene las actualizaciones de seguridad de Mandrake 8.2, haciéndose de esta manera:

[root@deneb root]#urpmi.addmedia updates \
ftp://ftp.sunet.se/pub/Linux/distributions/mandrake/updates/8.2/RPMS \
with ../base/hdlist.cz
added medium updates
retrieving description file of "updates"...
...retrieving done
retrieving source hdlist (or synthesis) of "updates"...
% Total
% Received % Xferd Average Speed
Time
Curr.

Dload Upload Total
Current Left
Speed
100 402k 100 402k
0
0
3653
0 0:01:52 0:01:52 0:00:00 4833
...retrieving done
examining whole urpmi database
[... listing trimmed ...]

Una vez la fuente ha sido añadida, se puede comprobar e instalar actualizaciones de seguridad con urpmi.update -a seguido de urpmi –auto-select.

Equivalencia de comandos.

Para los usuarios de Debian, esta equivalencia "grosera" puede ser útil cuando utilice las herramientas urpmi

Tabla. Correspondencias APT-URPMI

APT	URPMI
apt-get install	urpmi
apt-get upgrade	urpmi
apt-get update	urpmi.update -a
apt-get remove	urpme
apt-cache search	urpmf

Bueno ya esta bien de urmpi vamos con el servidor de correo sendmail bajo Mandrake. El mismo se encuentra en el CD1.

Con el CD1 instalado en el soporte correspondiente realizamos:

[root@localhost orlando]# cd /mnt/cdrom/Mandrake/RPMS
[4]

[root@localhost RPMS]# urpmi sendmail-8.12.8-1mdk.i586.rpm
Para resolver las dependencias, se instalarán los paquetes siguientes (3 MB):
cyrus-sasl-2.1.12-1mdk.i586
libdb4.0-4.0.14-6mdk.i586
libsasl2-2.1.12-1mdk.i586
sendmail-8.12.8-1mdk.i586
¿Está todo bien? (S/n) s
instalando /var/cache/urpmi/rpms/libsasl2-2.1.12-1mdk.i586.rpm /var/cache/urpmi/rpms/libdb4.0-4.0.14-6mdk.i586.rpm /var/cache/urpmi/rpms/cyrus-sasl-2.1.12-1mdk.i586.rpm sendmail-8.12.8-1mdk.i586.rpm
Preparando... ##
1:libsasl2 ##
2:libdb4.0 ##
3:cyrus-sasl ##
4:sendmail ##
[root@localhost RPMS]#

Un poco de paciencia porque tarda un poquito. Ya esta instalado. Pero un momento no sacad el CDROM1 todavía. Ahora vamos a instalar el preprocesador de macros m4.

[root@localhost RPMS]# urpmi m4-1.4ppre2-3mdk.i586.rpm
instalando m4-1.4ppre2-3mdk.i586.rpm
Preparando... urpmi m4-1.4ppre2-3mdk.i586.rpm
instalando m4-1.4ppre2-3mdk.i586.rpm
Preparando... ##
1:m4 ##
[root@localhost RPMS]#

[4] suponemos que el CDROM1 se encuentra en esa unidad, en cualquier caso en la que tengaís vosotros.

[root@localhost RPMS]#cd /usr
esto es para desmontar la unidad de CDROM
[root@localhost usr]# umount /mnt/cdrom
Ahora cambiamos de CD e instalaremos sendmail-cf, este se encuentra en el CDROM3.
[root@localhost usr]# mount /mnt/cdrom
[root@localhost usr]# cd /mnt/cdrom/Mandrake/RPMS3/
[root@localhost RPMS3]# urpmi sendmail-cf-8.12.8-1mdk.i586.rpm
instalando sendmail-cf-8.12.8-1mdk.i586.rpm
Preparando... ##
1:sendmail-cf ##
[root@localhost RPMS3]#
Pasemos con la configuración de Mandrake:

Para que el demonio arranque automaticamente cada vez que se inicie Mandrake debemos realizar lo siguiente:

[root@localhost orlando]#update sendmail defaults

En el directorio /usr/share/sendmail-cf/cf se incluyen varios ejemplos de configuración del fichero sendmail.mc. De hecho el que se usa en la mayoría de los casos es el que aparece como mandrake.mc

En el apartado 1.1.1.2 Configuración de Sendmail (apartado de guadalinex) se comenta el fichero sendmail.mc. Para adaptar el fichero a tus necesidades consulta este apartado ya que la configuración del fichero es independiente de la distribución que se use. Como se indica en ese apartado lo apropiado es modificar el fichero sendmail.mc y una vez lo tengamos configurado según nuestras necesidades esto cambia un poco con respecto a Debian debemos desde el directorio

[root@localhost cf]# **cd /usr/share/sendmail-cf/cf**

Realizar la modificación del fichero mc correspondiente. Una vez que se haya finalizado con las modificaciones desde ese directorio se debe de realizar:

[root@localhost cf]# **make mandrake.cf**
[5]

rm -f mandrake.cf
m4 ../m4/cf.m4 mandrake.mc >mandrake.cf || (rm -fmandrake.cf && exit 1)
chmod 444 mandrake.cf

Otra opción para conseguir el fichero es realizar la compilacion del mismo con el m4, esto sería de la forma siguiente:

[root@localhost cf]# m4 ../m4/cf.m4 mandrake.mc >sendmail.cf
[root@localhost cf]#

Esta última opción es independiente de la distribución.

[5] Donde el fichero original se supone que es mandrake.mc. Es decir el mismo nombre con extensión mc.

Capítulo 4

LA ALTERNATIVA : qmail. PARA QUE SIRVE. INSTALACIÓN. CONFIGURACIÓN.

qmail es un Agente de Transporte de Correo (MTA) para sistemas operativos tipo UNIX. Se trata de un sustituto completo para el sistema sendmail que se suministra con los sistemas operativos UNIX. qmail utiliza el Simple Mail Transfer Protocol (SMTP, Protocolo Simple de Transferencia de Correo) para intercambiar mensajes con los MTA (Agentes de Transporte de Correo) de otros sistemas. Atención: Su nombre es «qmail», no «Qmail».

Algunas de las ventajas de qmail sobre los MTA suministrados con el sistema[1] son:

Seguridad

Rendimiento[2]

Fiabilidad

Una vez que qmail ha aceptado un mensaje, garantiza que no se perderá. qmail soporta también un nuevo formato de bandeja de correo que funciona con seguridad incluso en NFS sin recurrir al bloqueo de ficheros.

Simplicidad

¿Cuando surge qmail?

qmail está desarrollado por Dan Bernstein (DJB), http://pobox.com/~djb/djb.html, un profesor de matemáticas ahora en la University of Illinois en Chicago. El Dr. Bernstein es asimismo bien conocido por su trabajo en el campo de la criptografía y por su juicio contra el gobierno de EE.UU. con respecto a la publicación de código fuente relativo a encriptación. Véase http://www.news.com/News/Item/0,4,36217,00.html?owv para más información sobre el proceso judicial.

El primer lanzamiento público de qmail, versión beta 0.70, tuvo lugar el 24 de enero de 1996. La primera versión gamma, 0.90, se lanzó el 1 de agosto de 1996.

La versión 1.0, el primer lanzamiento general, se anunció el 20 de febrero de 1997. La versión actual, 1.03, se lanzó el 15 de Junio de 1998.

4.1. ¿Cuánta gente usa qmail?

qmail es utilizado en cientos de Proveedores de Acceso a Internet, y miles de otros servidores. Algunos de ellos de alto nivel:

* ONElist, http://www.onelist.com gestiona la entrega de millones de mensajes diariamente, usando qmail.

[1] Sendmail, Postfix,....

[2] qmail paraleliza el envío de correo, llevando a cabo de forma predeterminada hasta 20 entregas simultáneas de correo.

* El mayor servidor de listas de correo LISTSERV no-LSTMP, listserv.acsu.buffalo.edu, ha estado en funcionamiento desde noviembre de 1996. Ha entregado doscientos millones de mensajes desde entonces.

* Hotmail, http://www.hotmail.com, con treinta millones de usuarios, ha estado utilizando qmail para el envío de correo saliente desde 1997.[3]

Éstas son las 10 razones principales para usar qmail.

1. Seguridad, qmail no deja a los intrusos apoderarse de su máquina.

2. Fiabilidad. qmail nunca pierde correo.

3. Velocidad. qmail entrega su correo mucho más rápido que sendmail, sin comprometer la fiabilidad.

4. Bajo consumo de memoria. qmail puede con docenas de entregas simultáneas en ese viejo 486 de 16 MB.

5. Listas de correo gestionadas por usuarios. Los usuarios no tienen que incordiar al administrador del sistema para crear nuevas listas.

6. Dominios virtuales de forma fácil. qmail fue el pionero en el soporte a múltiples dominios.

7. Administración sencilla y directa. qmail funciona con mínimas complicaciones.

8. Flexibilidad en cuanto a los programas utilizados para entrega. qmail proporciona una potente interfaz para ser usada con procesadores de correo externos.

9. Rutas de Retorno para Envolturas Variables. Esta prestación (utilizada por ezmlm), permite una gestión de mensajes devueltos automática al 100 % de listas de distribución de correo de cualquier tamaño.

10. El formato maildir. Esta característica facilita preparar servidores POP distribuidos de gran volumen.

4.2. qmail es seguro

Existe en la actualidad una recompensa de US$1000 a quien se meta en una maquina rompiendo el servicio de email por qmail, recompensa que hasta la fecha nadie ha reclamado. En realidad la recompensa fue inicialmente de $500, y la ofrecía el propio autor. La retiró al año, y luego otros ofrecieron los $1000 ¿Es pues seguro? Juzgazlo vosostros mismos.

4.3. Arquitectura de qmail

qmail está compuesto por una serie de programas (módulos) que llevan a cabo tareas diferenciadas.

4.4. Licencia

El copyright de qmail pertenece a su autor, Dan Bernstein, y no se distribuye con una declaración de derechos del usuario. Se concede el derecho de distribuir el código fuente de qmail. La letra pequeña es que puede usar qmail con cualquier finalidad, y puede redistribuir libremente distribuciones de código fuente de qmail pero sin modificaciones, puede certificar distribuciones binarias var-qmail, y puede redistribuir parches para qmail. Pero no puede distribuir código fuente de qmail modificado o distribuciones de binarios que no sean var-qmail.

4.4.1. Instalacion y experiencias.

Antes de empezar la instalación pasaremos por indicar que la instalación de este servidor es quizas una de las complicadas, con esto no deseamos desanimar a nadie pero si indicaros que quizas todo no salga a la primera y que hay que tener un poco de paciencia, y que hay que escribir bastante más que en los otros dos servidores.

¿No se distingue la instalación? Pues no, en ambos casos se utilizará la misma instalacion ya que se tratara de compilar un codigo fuente.

[3] Datos de la pagina original de qmail: “Según he sido informado, tras la compra de Hotmail por parte de Microsoft, intentaron migrar a Microsoft Exchange bajo Windows NT. Exchange reventó” cita de DJB

En este manual vamos a aprender como montar un servidor de correo paso a paso, intentando aprender de camino para que sirve cada paso que damos. Lo primero es conocer un conjunto de aplicaciones:

1.- ¿Qué es qmail?

qmail es un Agente de Transporte de Correo (MTA, Mail Transport Agent en inglés) para sistemas operativos tipo UNIX. Se trata de un sustituto completo para el sistema sendmail que se suministra con los sistemas operativos UNIX. qmail utiliza el Simple Mail Transfer Protocol (SMTP, Protocolo Simple de Transferencia de Correo) para intercambiar mensajes con los MTA (Agentes de Transporte de Correo) de otros sistemas.

2.- ¿Qué es smtp-auth?

smtp-auth es un parche para qmail, que activa el soporte para el protocolo de autentificación SMTP con la búsqueda de diferentes tipos de autentificación como: LOGIN, PLAIN y CRAM-MD5. Este parche nos va a ser útil ya que nos previene de la posibilidad que nuestro servidor sirva para hacer spam.

3.- ¿Qué es daemontools?

daemontools es una colección de utilidades para el manejo de servicios UNIX.

4.- ¿Qué es vpopmail?

La manipulación de dominios virtuales es una edición común planteada por los nuevos usuarios en las comunidades de qmail y del postfix. Inter7 ha desarrollado el vpopmail (vchkpw), una paquete de software libre del GLP, para proporcionar una manera fácil de manejar dominios virtuales del email y cuentas del email y no /etc/passwd en su qmail o postfix. Además es muy útil ya que podemos tener varios dominios en una sola dirección IP.

Una vez visto por encima que es cada cosa que vamos a usar nos disponemos a empezar con la instalación de nuestro servidor de correo, la misma es adaptable para cualquier distribución, ya que lo unico que se requiere es que se tenga un poco de conocimiento con los comandos, y el

4.4.1.1. - Requisitos

Debeís tener cuenta que tiene que estar desinstalado el MTA, que puede ser : sendmail o exim,postfix... O cualquier otro.
[4]

Otro requisito previo es instalar un compilador de C, para no entrar en demasiados detalles técnicos deberiamos de probarlo escribiendo esto:

[orlando@orlando borrame]$ cc

cc: no input files

o bien gcc debiendo obtener el mismo mensaje.[5]

En el caso de obtener otro mensaje deberiamos de instalarlo.

El último requisito es que si anteriormente habiamos instalado estos paquetes (necesarios para probar sendmail) no hace falta realizar esta instalacion, si no ha sido así debemos instalar soporte de pop e imap

En Guadalinex:

apt-get install uw-imapd ipopd

En Mandrake 9.1:

urpmi imap-2002a-2mdk

En Mandrake 10:

urpmi imap-2002d-8mdk

En cualquiera de los casos debemos de instalar aquellas librerias o paquetes suplementarios que se nos solicite.

4.4.1.2. Comenzamos la instalación

Una vez aqui damos por supuesto que se cumplen los requisitos anteriores.

Vamos a descargar los paquetes necesarios para la instalación de nuestro servidor de correos. Por comodidad como la mayoría de los comandos que vamos a utilizar son en modo consola por lo que vamos a utilizar wget para descargarnos programas.

[4] Si se ha instalado el programa sendmail se debe desinstalar previamente a realizar la instalación de qmail.

[5] en principio daremos por supuesto el compilador.

Empezamos por descargar todos los paquetes necesarios.

Indicar que todos los comanados que se enuncian a continuación deben de realizarse como root.

mkdir /usr/local/src/qmail

cd /usr/local/src/qmail

wget http://cr.yp.to/daemontools/daemontools-0.76.tar.gz

wget http://cr.yp.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz

wget http://www.qmail.org/netqmail-1.05.tar.gz

wget http://members.elysium.pl/brush/qmail-smtpd-auth/dist/qmail-smtpd-auth-0.31.tar.gz

wget http://www.inter7.com/devel/vpopmail-5.3.14.tar.gz

wget http://www.qmail.org/qmailqueue-patch

Descomprimir paquetes

Ahora vamos a proceder a descomprimir cada paquete descargado en el sitio que le corresponde.

cd /usr/local/src/qmail

tar -xvzf netqmail-1.05.tar.gz

tar -xzvf ucspi-tcp-0.88.tar.gz

tar -xzvf qmail-smtpd-auth-0.31.tar.gz

tar -xzvf vpopmail-5.3.14.tar.gz

tar -xzf /usr/local/src/qmail/daemontools-0.76.tar.gz

Aplicación de parches a qmail

Una vez descomprimido cada paquete vamos a proceder a aplicar los parches pertinentes a qmail, el paquete netqmail-1.05 contiene una serie de parches, ademas del propio qmail. Es necesario aplicar estos parches antes de empezar la compilación y ahora es un buen momento, antes de que se nos olvide. Para ello, únicamente es necesario ejecutar el fichero collate.sh :

cd netqmail-1.05

[root@orlando qmail]# cd /usr/local/src/netqmail-1.05

[root@orlando netqmail-1.05]# ./collate.sh

You should see 7 lines of text below. If you see anything

else, then something might be wrong.

[1] Extracting qmail-1.03...

[2] Patching qmail-1.03 into netqmail-1.05. Look for errors below:

24

[4] The previous line should say 24 if you used GNU patch.

[5] Renaming qmail-1.03 to netqmail-1.05...

[6] Continue installing qmail using the instructions found at:

[7] http://www.lifewithqmail.org/lwq.html#installation

Este parche nos sirve para poder usar smtp-auth.

cd /usr/local/src/qmail/qmail-smtpd-auth-0.31

[root@portatil qmail-smtpd-auth-0.31]# cp README.auth base64.cbase64.h ../netqmail-1.05/netqmail-1.05

[root@portatil qmail-smtpd-auth-0.31]# patch -d ../netqmail-1.05/netqmail-1.05<auth.patch

[root@portatil qmail-smtpd-auth-0.31]# patch -d ../netqmail-1.05/netqmail-1.05 <auth.patch

patching file Makefile

patching file TARGETS

patching file qmail-smtpd.8

patching file qmail-smtpd.c

Hunk #2 succeeded at 62 with fuzz 1.

Hunk #3 succeeded at 241 with fuzz 1.

[root@portatil qmail-smtpd-auth-0.31]#

Instalación de qmail

Bueno ya hemos aplicado los parches que necesitabamos, ahora nos toca instalar paso a paso nuestro qmail.

mkdir /var/qmail

mkdir /var/log/qmail

groupadd nofiles

useradd -g nofiles -d /var/qmail/alias alias
useradd -g nofiles -d /var/qmail qmaild
useradd -g nofiles -d /var/qmail qmaill
useradd -g nofiles -d /var/qmail qmailp
groupadd qmail
useradd -g qmail -d /var/qmail qmailq
useradd -g qmail -d /var/qmail qmailr
useradd -g qmail -d /var/qmail qmails
cd /usr/local/src/qmail/netqmail-1.05/netqmail-1.05
y luego escribir
make setup check
Ahora empieza el proceso de compilación.[6]
Si todo va bien debemos tener en la pantalla las dos últimas lineas con los mensajes:
./install
./instcheck

Después de que se complete el proceso de compilación (puede llevar varios minutos), es necesario realizar la configuración post-instalación. Hay un par de scripts que nos facilitarán esta labor:

./config En caso que tengamos el DNS configurado correctamente

./config-fast host.dominio En caso que no tengamos DNS configurado[7] Nosotros ejecutaremos config-fast para no tener que depender del DNS:

[root@portatil netqmail-1.05]# ./config-fast portatil.micentro.es
Your fully qualified host name is portatil.micentro.es.
Putting portatil.micentro.es into control/me...
Putting micentro.es into control/defaultdomain...
Putting micentro.es into control/plusdomain...
Putting portatil.micentro.es into control/locals...
Putting portatil.micentro.es into control/rcpthosts...
Now qmail will refuse to accept SMTP messages except toportatil.micentro.es.
Make sure to change rcpthosts if you add hosts to locals or virtualdomains!
[root@portatil netqmail-1.05]#

Con la ejecución de este script ya tendríamos qmail instalado en el sistema y listo para ejecutarse. A continuación veremos el proceso de instalación de los paquetes con las herramientas que mejoran qmail y posteriormente cómo arrancarlo y comprobar su funcionamiento.

Explicacion del comando
./config-fast maquina.dominio.com
maquina.dominio.com quiere decir que en maquina ponemos el nombre que le dimos a nuestro servidor y en dominio el dominio el cual hayamos contratado, un ejemplo sería: Mi maquina se llama mailhost y mi dominio contratado es www.pruebas.com, entonces mi nombre sería mailhost.pruebas.com

4.4.1.3. Instalación de ucspi-tcp (tcpserver)

No hemos hablado anteriormente de este apartado ya que este paquete tan solo contiene tcpserver y tcpclient, las herramientas para construir la linea de comandos del servidor cliente.

qmail precisa de un mecanismo para que el demonio qmail-smtpd se lance cada vez que llega un intento de conexión por el puerto SMTP desde el exterior del servidor. Los creadores de qmail recomiendan el uso del programa tcpserver que está disponible como parte del paquete ucspi-tcp.

Para instalar ucspi-tcp es necesario situarnos en el directorio
cd /usr/local/src/qmail/ucspi-tcp-0.88
y ejecutar lo siguiente:
patch </usr/local/src/qmail/netqmail-1.05/other-patches/ucspi-tcp-0.88.errno.patch
make

[6] para algunos empezaran a salir numeros y letras en la pantalla. Todo va bien.
[7] Este será el que utilicemos en la mayoria de los casos.

make setup check
Ya tendríamos instalado ucspi-tcp-0.88.

4.4.1.4. Instalación de daemontools

cd /usr/local/src/qmail/admin/daemontools-0.76/
cd daemontools-0.76/
cd src/
patch </usr/local/src/qmail/netqmail-1.05/other-patches/daemontools-0.76.errno.patch
cd ..
package/install
Ya tendríamos instalado daemontools-0.76.
Para comprobar que las daemontools estan arrancadas, hacer un ps ax y mirar si aparece "/bin/sh /command/svscanboot" y "svscan /service".

4.4.1.5. Instalación de vpopmail.

Bueno ya vamos avanzando y cada vez queda menos para terminar nuestro servidor de correo. Ahora nos disponemos a instalar vpopmail.
groupadd -g 89 vchkpw
useradd -g vchkpw -u 89 vpopmail
Una vez añadido el grupo de vpopmail y el usuario nos disponemos a compilar vpopmail, todo ello como root.
cd /usr/local/src/qmail/vpopmail-5.3.14
./configure –enable-ip-alias-domains=y –enable-ucspi-dir=../ucspi-tcp-0.88 –enable-logging=y –enable-tcpserver-file=/etc/tcp.smtp –enable-many-domains=y –enable-passwd=y
make
make install-strip
Agregamos nuestro dominio y un usuario de prueba, para que una vez hechos los scripts los servicios abran los puertos pertinentes.
/home/vpopmail/bin/vadddomain pruebas.com <contraseña>
/home/vpopmail/bin/vadduser usuario@pruebas.com <contraseña>
Ya finalizamos la instalación pero ahora tenemos que configurar los scripts.

4.4.1.6. Configuracion y scripts

Ahora comienza la parte de configuración.
En el directorio /var/qmail/boot hay varios scripts de ejemplo que arrancan qmail para diferentes combinaciones. Sin embargo, nosotros usaremos el siguiente script /var/qmail/rc
#!/bin/sh
exec env - PATH="/var/qmail/bin:$PATH" \
qmail-start "‘cat /var/qmail/control/defaultdelivery‘"
Una vez creado con un editor se ejecuta lo siguiente:
chmod 755 /var/qmail/rc
Lo siguiente es decidir el modo de entrega para aquellos mensajes que no son entregados.
echo ./Mailbox >/var/qmail/control/defaultdelivery
Si se ejecuta el script /var/qmail/rc, qmail se arrancará de forma parcial. Para que qmail se arranque de forma automática cada vez que el sistema es arrancado, y que pare cuando queramos apagar el sistema de forma ordenada, usaremos el script /var/qmail/bin/qmailctl. Lo crearemos mediante un editor de texto con el contenido que se expondrá a continuación .En esta parte tenemos dos posibilidades para el script: la primera escribir el script tal y como lo encontraremos a continuación
Script de inicio de qmail (/var/qmail/rc)
vi /var/qmail/rc
#!/bin/sh

```
# Using stdout for logging
# Using control/defaultdelivery from qmail-local to deliver messages by default
exec env - PATH="/var/qmail/bin:$PATH" \
qmail-start "‘cat /var/qmail/control/defaultdelivery‘"
chmod 755 /var/qmail/rc
echo ./Maildir/ >/var/qmail/control/defaultdelivery
```

Automatización del script de inicio (/var/qmail/bin/qmailctl)

```
vi /var/qmail/bin/qmailctl
#!/bin/sh
# For Red Hat chkconfig
# chkconfig: - 80 30
# description: the qmail MTA
PATH=/var/qmail/bin:/bin:/usr/bin:/usr/local/bin:/usr/local/sbin
export PATH
QMAILDUID=‘id -u qmaild‘
NOFILESGID=‘id -g qmaild‘
case "$1" in
start)
echo "Starting qmail"
if svok /service/qmail-send ; then
svc -u /service/qmail-send
else
echo qmail-send supervise not running
fi
if svok /service/qmail-smtpd ; then
svc -u /service/qmail-smtpd
else
echo qmail-smtpd supervise not running
fi
if [ -d /var/lock/subsys ]; then
touch /var/lock/subsys/qmail
fi
if svok /service/qmail-pop3d ; then
svc -u /service/qmail-pop3d
else
echo qmail-pop3d supervise not running
fi
;;
stop)
echo "Stopping qmail..."
echo " qmail-smtpd"
svc -d /service/qmail-smtpd
echo " qmail-send"
svc -d /service/qmail-send
if [ -f /var/lock/subsys/qmail ]; then
rm /var/lock/subsys/qmail
fi
echo " qmail-pop3d"
svc -d /service/qmail-pop3d
;;
stat)
svstat /service/qmail-send
svstat /service/qmail-send/log
svstat /service/qmail-smtpd
```

```
svstat /service/qmail-smtpd/log
svstat /service/qmail-pop3d
svstat /service/qmail-pop3d/log
qmail-qstat
;;
doqueue|alrm|flush)
echo "Flushing timeout table and sending ALRM signal to qmail-send."
/var/qmail/bin/qmail-tcpok
svc -a /service/qmail-send
;;
queue)
qmail-qstat
qmail-qread
;;
reload|hup)
echo "Sending HUP signal to qmail-send."
svc -h /service/qmail-send
;;
pause)
echo "Pausing qmail-send"
svc -p /service/qmail-send
echo "Pausing qmail-smtpd"
svc -p /service/qmail-smtpd
echo "Pausing qmail-pop3d"
svc -p /service/qmail-pop3d
;;
cont)
echo "Continuing qmail-send"
svc -c /service/qmail-send
echo "Continuing qmail-smtpd"
svc -c /service/qmail-smtpd
echo "Continuing qmail-pop3d"
svc -c /service/qmail-pop3d
;;
restart)
echo "Restarting qmail:"
echo "* Stopping qmail-smtpd."
svc -d /service/qmail-smtpd
echo "* Sending qmail-send SIGTERM and restarting."
svc -t /service/qmail-send
echo "* Restarting qmail-smtpd."
svc -u /service/qmail-smtpd
echo "* Restarting qmail-pop3d."
svc -t /service/qmail-pop3d
;;
cdb)
tcprules /etc/tcp.smtp.cdb /etc/tcp.smtp.tmp </etc/tcp.smtp
chmod 644 /etc/tcp.smtp.cdb
echo "Reloaded /etc/tcp.smtp."
;;
help)
cat <<HELP
stop – stops mail service (smtp connections refused, nothing goes out)
start – starts mail service (smtp connection accepted, mail can go out)
```

pause – temporarily stops mail service (connections accepted, nothing leaves)
cont – continues paused mail service
stat – displays status of mail service
cdb – rebuild the tcpserver cdb file for smtp
restart – stops and restarts smtp, sends qmail-send a TERM & restarts it
doqueue – schedules queued messages for immediate delivery
reload – sends qmail-send HUP, rereading locals and virtualdomains
queue – shows status of queue
alrm – same as doqueue
flush – same as doqueue
hup – same as reload
HELP
;;
*)
echo "Usage: $0 {start|stop|restart|doqueue|flush|reload|stat|pause|cont|cdb|queue|help}"
exit 1
;;
esac
exit 0

Para ahorrarnos el teclear todo el script podemos bajarlo de http://www.lifewithqmail.org/qmailctl-script-dt70

Una vez creado lo hacemos ejecutable y accesible en /usr/bin:
[root@portatil root]# chmod 755 /var/qmail/bin/qmailctl
[root@portatil root]# ln -s /var/qmail/bin/qmailctl /usr/bin
Ahora nos disponemos a crear los enlaces de cada inicio en cada capa y los directorios de supervise.
ln -s /var/qmail/bin/qmailctl /etc/init.d/qmail
ln -s /etc/init.d/qmail /etc/rc0.d/K30qmail
ln -s /etc/init.d/qmail /etc/rc1.d/K30qmail
ln -s /etc/init.d/qmail /etc/rc2.d/S80qmail
ln -s /etc/init.d/qmail /etc/rc3.d/S80qmail
ln -s /etc/init.d/qmail /etc/rc4.d/S80qmail
ln -s /etc/init.d/qmail /etc/rc5.d/S80qmail
ln -s /etc/init.d/qmail /etc/rc6.d/K30qmail
mkdir -p /var/qmail/supervise/qmail-send/log
mkdir -p /var/qmail/supervise/qmail-smtpd/log
mkdir -p /var/qmail/supervise/qmail-pop3d/log
mkdir -p /var/log/qmail/smtpd
mkdir -p /var/log/qmail/pop3d
chown qmaill /var/log/qmail /var/log/qmail/smtpd /var/log/qmail/pop3d

Scripts de arranque y logeo de qmail-send

vi /var/qmail/supervise/qmail-send/run
#!/bin/sh
exec /var/qmail/rc
este es uno
despues escribir
chmod 755 /var/qmail/supervise/qmail-send/run
vi /var/qmail/supervise/qmail-send/log/run
#!/bin/sh
exec /usr/local/bin/setuidgid qmail /usr/local/bin/multilog t /var/log/qmail
despues escribir
chmod 755 /var/qmail/supervise/qmail-send/log/run

Scripts de arranque y logeo de qmail-smtpd (smtp-auth)

Para poder usar smtp-auth, vamos a crear una copia de vchkpw, aquí hay varios puntos importantes que hay que tener en cuenta.

cp /home/vpopmail/bin/vchkpw /home/vpopmail/bin/vchkpw.smtp
cd /home/vpopmail/bin/
chown vpopmail.vchkpw /home/vpopmail/bin/vchkpw.smtp
chmod +s /home/vpopmail/bin/vchkpw.smtp

Estas líneas anteriores eran para mantener una copia intacta de vchkpw, ahora vamos a proceder a crear los scripts anteriormente citados.

vi /var/qmail/supervise/qmail-smtpd/run
#!/bin/sh
QMAILDUID=‘id -u qmaild‘
NOFILESGID=‘id -g qmaild‘
MAXSMTPD=‘cat /var/qmail/control/concurrencyincoming‘
exec /usr/local/bin/softlimit -m 4000000 /usr/local/bin/tcpserver -H -R -l 0 -c "$MAXSMTPD" -x /etc/tcp.smtp.cdb -u "$QMAILDUID" -g "$NOFILESGID" 0 smtp /var/qmail/bin/qmail-smtpd dominio.com /home/vpopmail/bin/vchkpw.smtp /bin/true 2>&1

NOTA: dominio.com lo tendremos que sustituir por el nombre de dominio asignado, en este caso era pruebas.com, pero hay que observar que tan solo ponemos pruebas.com y no ponemos mail-host.pruebas.com

chmod 755 /var/qmail/supervise/qmail-smtpd/run

Tenemos que crear el archivo concurrencyincoming, el cual estará determinado por un número concurrente para realizar envíos simultáneos.

echo 20 >/var/qmail/control/concurrencyincoming
chmod 644 /var/qmail/control/concurrencyincoming

Ahora vamos a crear el script de logeo de qmail-smtpd.

vi /var/qmail/supervise/qmail-smtpd/log/run
#!/bin/sh
exec /usr/local/bin/setuidgid qmail /usr/local/bin/multilog t /var/log/qmail/smtpd

Scripts de arranque y logeo de qmail-pop3d

vi /var/qmail/supervise/qmail-pop3d/run
#!/bin/sh
exec /usr/local/bin/tcpserver -v -R -H -l 0 -u 89 -g 89 0 110 /var/qmail/bin/qmail-popup dominio.com /home/vpopmail/bin/vchkpw /var/qmail/bin/qmail-pop3d Maildir 2>&1

NOTA: dominio.com lo tendremos que sustituir por el nombre de dominio asignado, en este caso era pruebas.com, pero hay que observar que tan solo ponemos pruebas.com y no ponemos mailhost.pruebas.com

vi /var/qmail/supervise/qmail-pop3d/log/run
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog t /var/log/qmail/pop3d

Ahora nos disponemos a hacer ejecutables todos estos archivos y enlazarlos con daemontools.

chmod 755 /var/qmail/supervise/qmail-send/run
chmod 755 /var/qmail/supervise/qmail-send/log/run
chmod 755 /var/qmail/supervise/qmail-smtpd/run
chmod 755 /var/qmail/supervise/qmail-smtpd/log/run
chmod 755 /var/qmail/supervise/qmail-pop3d/run
chmod 755 /var/qmail/supervise/qmail-pop3d/log/run
ln -s /var/qmail/supervise/qmail-send /var/qmail/supervise/qmail-smtpd /service
ln -s /var/qmail/supervise/qmail-pop3d /service

Una vez hecho este último paso nuestro servidor de correo debería estar funcionando, una forma de comprobarlo es de la siguiente forma.

telnet localhost 25
telnet localhost 110

También deberemos probar el script que creamos (qmailctl) para parar los servicios y volver a iniciarlos con este comando.

/etc/init.d/qmail stop
/etc/init.d/qmail start

Esto consiste en activar la ip de nuestro servidor para que él sea el único que este autorizado para mandar correo sin autentificarse, es decir, que para poder mandar correo nos pedirá una contraseña, si nuestra ip estuviese dentro de este relaying no nos pediría dicha contraseña.

echo ?127.0.0.1:allow,RELAYCLIENT=??? > >/etc/tcp.smtp

qmailctl cdb

Una vez añadida la ip de localhost, nos disponemos a añadir una línea en el crontab, para que cada 10 minutos se borre la lista virtual de ip's.

crontab ?e

9-59,10 * * * * /home/vpopmail/bin/clearopensmtp 2>&1 >/dev/null

Para comprobar que hemos añadido correctamente la línea en nuestro crontab ejecutaremos el siguiente comando:

crontab ?l

Alias básicos

Por último vamos a añadir unos alias básicos que necesitamos para el correcto funcionamiento de qmail.

ln -s /var/qmail/bin/sendmail /usr/lib

ln -s /var/qmail/bin/sendmail /usr/sbin

echo user >/var/qmail/alias/.qmail-root

echo user >/var/qmail/alias/.qmail-postmaster

ln -s .qmail-postmaster /var/qmail/alias/.qmail-mailer-daemon

chmod 644 /var/qmail/alias/.qmail-root /var/qmail/alias/.qmail-postmaster

Con esto ya esta todo

Ahora debemos tener qmail ejecutándose en el sistema. Antes de nada, ejecutar qmailctl start para verificar los servicios que están activos.

[root@portatil root]# qmailctl start

/service/qmail-send: up (pid 7530) 20 seconds

/service/qmail-send/log: up (pid 7528) 20 seconds

/service/qmail-smtpd: up (pid 7553) 17 seconds

/service/qmail-smtpd/log: up (pid 7535) 19 seconds

messages in queue: 0

messages in queue but not yet preprocessed: 0

Los cuatro servicios que aparecen, deben estar en estado up por más de 1 segundo. En caso que no sea así, probablemente hay algún error en la ejecución de alguno de los scripts o se ha saltado alguno de los pasos que hemos descrito con anterioridad. Será necesario volver atrás y verificar detenidamente todos los pasos.

Existe una forma más sencilla para comprobar la instalación. Podemos utilizar el fichero que se encuentra en http://lifewithqmail.org/inst_check, el cual realizará un chequeo de forma automática.

Capítulo 5

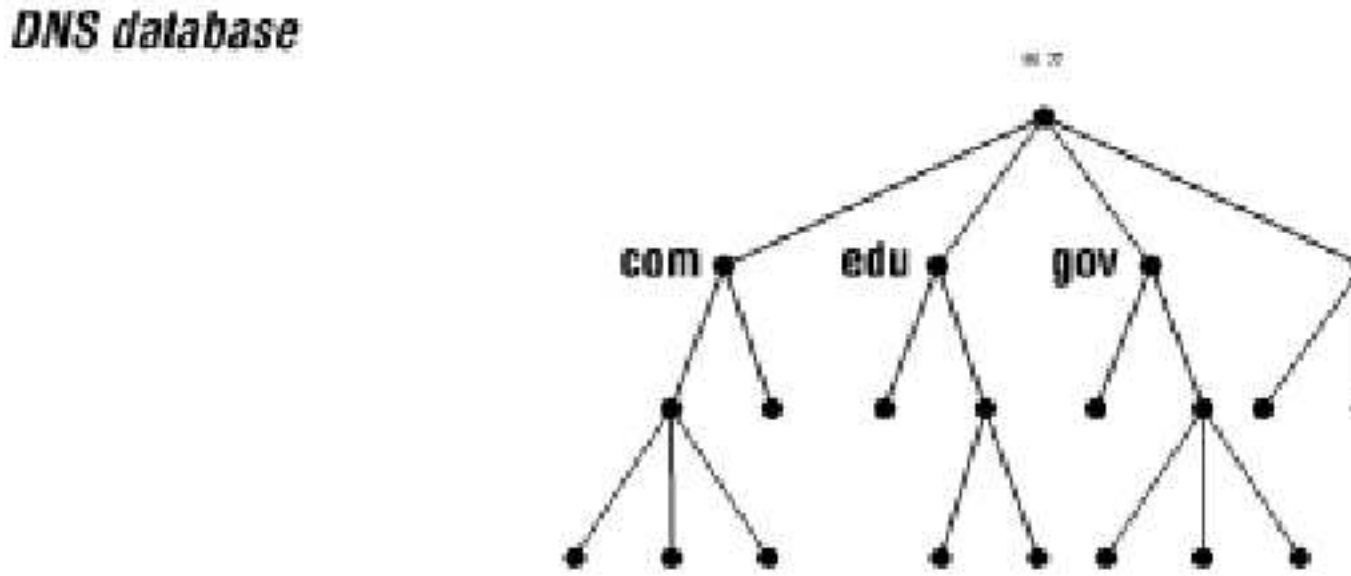
Servidor de nombres DNS

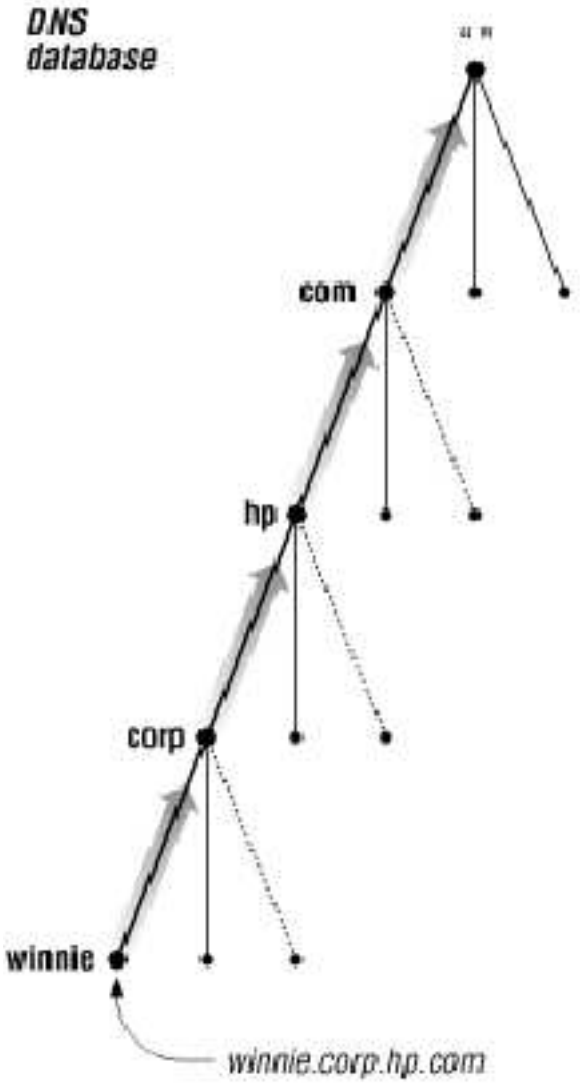
Puede que no conozcas mucho sobre el Sistema de Nombres de Dominio (DNS) -todavía- pero siempre que usas Internet, estás usando el DNS. Cada vez que envías un correo electrónico o navegas por la Web, dependes del DNS. (DNS and Bind, PAUL ALBITZ y CRICKET LIU)

5.1. Estructura del DNS.

Llegó la hora de las direcciones simbólicas. Las direcciones IP han campado a sus anchas y la verdad es que para nosotros son difíciles de recordar y propensas a errores. Donde esté un nombre simple y descriptivo como thales.cica.es, que se quiten todas las direcciones IP como su equivalente 172.26.0.2 ¿o era 150.214.22.12? ¡Ah! no, es 150.214.5.10. Véis, nuestra capacidad simbólica es superior a nuestra capacidad de recordar números.

El sistema DNS es una base de datos distribuida. Presenta una jerarquía en la que su parte más alta es el ”punto” o raíz y de él cuelgan los dominios de primer nivel (.com, .edu, .es, etc).





Su lectura en el orden jerárquico se realiza de derecha a izquierda. Por ejemplo, para la máquina thales.cica.es, primero en la jerarquía se encuentra el dominio de primer nivel (.es), luego va el subdominio o subdominios (en este caso, cica) y por último el nombre de la máquina (thales).

En la figura, podemos ver cómo sería la estructura jerárquica para la máquina winnie.corp.hp.com.

Los dominios genéricos de primer nivel son los .com, .edu, .org... más los correspondientes a los países (.es, .it, .uk, .pt,...). En Noviembre de 2000, ICANN (Internet Corporation for Assigned Names and Numbers www.icann.org) anunció la aparición de 7 nuevos dominios de primer nivel: .biz, .info, .name, .pro, .aero, .coop y .museum.

Además de estar jerarquizada, esta estructura se encuentra delegada. Veamos qué significa esto aplicándolo a nuestra dirección thales.cica.es.

ICANN es una organización sin fines de lucro que opera a nivel internacional, responsable de asignar espacio de direcciones numéricas de protocolo de Internet (IP), identificadores de protocolo y de las funciones de gestión del sistema de nombres de dominio de primer nivel genéricos (gTLD) y de códigos de países (ccTLD).

El dominio de primer nivel .es se encuentra delegado por ICANN a España, más concretamente al Organismo Red.es[1]. A su vez, Red.es delega la administración del subdominio cica al Centro Informático

[1] Anteriormente era Rediris la encargada, a través del ES-NIC.

Científico de Andalucía, que se convierte en responsable de todo lo que cuelgue de él, y así por ejemplo, puede darle nombre (y apellidos) a la máquina thales como thales.cica.es.

Este sistema hace que a pesar de la distribución y delegación de responsabilidades, todo funcione con la necesaria coordinación a nivel regional y mundial.

Para profundizar en el tema y conocer más sobre el dominio .es, podéis consultar en

http://plugindoc.mozdev.org/linux.html

Al principio, con pocas máquinas en Internet, bastaba para mantener este sistema con unos ficheros de nombre HOSTS.TXT en los que se encontraban los nombres de las máquinas uno a uno. A medida que el sistema fue creciendo, se hacía necesario el soporte de un sistema más potente, que es el basado en Servidores de Nombres.

5.2. ¿Qué necesito del DNS?

Ésta es una de las principales cuestiones a las que deberemos responder a la hora de configurar y gestionar nuestros sistemas.

La gran mayoría de vosotros, no necesitará montar y configurar un servidor de nombres, pero sí que los utilizaréis prácticamente en cada momento. Por ello, el comprender su funcionamiento y los recursos que ofrece es de gran ayuda.

Como vimos en la primera entrega, nuestra máquina Linux necesita saber cómo resolver las direcciones simbólicas a numéricas. Ello se hacía mediante los ficheros /etc/hosts, /etc/nsswitch.conf y /etc/resolv.conf, o los correspondientes interfaces gráficos.

Debemos diferenciar la utilización que hacemos de los servidores de nombres, del hecho de montar un servidor de nombres propio. Es algo así como la diferencia entre utilizar un procesador de textos para nuestro trabajo diario y el desarrollar un procesador de textos nosotros mismos.

5.3. Recursos del Servidor de Nombres

Para ver qué nos ofrece un servidor de nombres utilizaremos la herramienta dig. En su forma más simple, le preguntamos como argumento con un nombre de host para conocer la dirección que le corresponde.

```
[root@nuevo pedro]# dig www.cica.es
; <<>>DiG 9.2.3 <<>>www.cica.es
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36848
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 0
;; QUESTION SECTION:
;www.cica.es. IN A
;; ANSWER SECTION:
www.cica.es. 2951 IN CNAME ataman.cica.es.
ataman.cica.es. 2951 IN A 150.214.5.119
;; AUTHORITY SECTION:
cica.es. 3600 IN NS sun.rediris.es.
cica.es. 3600 IN NS dns1.cica.es.
cica.es. 3600 IN NS dns2.cica.es.
cica.es. 3600 IN NS chico.rediris.es.
;; Query time: 4571 msec
;; SERVER: 192.168.2.2#53(192.168.2.2)
;; WHEN: Thu Jan 20 17:40:05 2005
;; MSG SIZE rcvd: 150
```

Esta es la salida del comando dig, bastante parlanchina, por cierto. La respuesta principal es la línea:

```
ataman.cica.es. 2951 IN A 150.214.5.119
```

que nos dice que la máquina `ataman.cica.es` tiene la dirección IP `150.214.5.119`. Además, nos dice que es una dirección de tipo INternet (IN) y es un recurso de tipo A (Address). El valor `2951` es un valor de tiempo de vida (ttl).

Pero, ¿si yo he preguntado por `www.cica.es?`.

La línea anterior define un alias o dirección símbolica:

```
www.cica.es. 2951 IN CNAME ataman.cica.es
```

Además, dentro de su cortesía nos regala información adicional, como las líneas

```
cica.es. 3600 IN NS sun.rediris.es.
```

que nos indican cuáles son los servidores de nombres "oficiales" para la zona cica.es, que son cuatro, con el tipo de recurso NS (Name Server).

Además el tiempo que ha tardado la consulta, a quién y cuándo. La siguiente línea

```
;; SERVER: 192.168.2.2#53(192.168.2.2)
```

nos dice que la consulta ha sido realizada al servidor con dirección IP `192.168.2.2` por el puerto 53, que es el que utiliza el servicio DNS. Como curiosidad, comentar que las consultas a los servidores DNS pueden realizarse tanto por TCP como por UDP.

El comando dig nos será de gran ayuda para consultar a los servidores de nombres. Una llamada típica al comando dig es de la forma:

```
dig @servidor_de_nombres recurso tipo_del recurso
```

donde:

servidor_de_nombres es el servidor de nombres al que vamos a preguntar. En caso de que no lo especifiquemos, preguntará a los servidores de nombres que estén en el fichero /etc/resolv.conf

recurso es el nombre o dirección del que queremos consultar información

tipo_del_recurso es el tipo del recurso que buscamos. Si no especificamos ninguno, buscará el tipo A por defecto.

Un servidor de nombres nos ofrece varios tipos de recursos. Veremos a continuación los más importantes.

A (Address) Nos da la correspondencia de dirección simbólica a dirección IP

CNAME (canonical name) Nos especifica un alias o apodo para una dirección simbólica

MX (mail exchanger) Indica la máquina o las máquinas que recibirán el correo

NS (name server) Indica los servidores de nombres oficiales para el dominio

PTR (pointer) Nos da la resolución inversa de una dirección IP a una dirección simbólica

SOA (start of authority) Autoridad sobre el Dominio de nombres.

Exprimamos un poco más el comando dig.

Le preguntaremos al servidor de nombres 150.214.4.343.6, que como vimos en el anterior comando, es un servidor de nombres oficial para el dominio cica.es.

```
[root@nuevo pedro]# dig @dns1.cica.es ANY cica.es
; <<>>DiG 9.2.3 <<>>@dns1.cica.es ANY cica.es
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29923
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 6
;; QUESTION SECTION:
;cica.es. IN ANY
;; ANSWER SECTION:
cica.es. 172800 IN SOA dns1.cica.es. hostmaster.cica.es
cica.es. 172800 IN NS chico.rediris.es.
cica.es. 172800 IN NS sun.rediris.es.
cica.es. 172800 IN NS dns1.cica.es.
cica.es. 172800 IN NS dns2.cica.es.
cica.es. 300 IN MX 10 smtp.cica.es.
cica.es. 300 IN MX 15 smtp2.cica.es.
;; ADDITIONAL SECTION:
```

```
sun.rediris.es. 161630 IN A 130.206.1.2
dns1.cica.es. 172800 IN A 150.214.5.83
dns2.cica.es. 172800 IN A 150.214.4.35
chico.rediris.es. 17630 IN A 130.206.1.3
smtp.cica.es. 172800 IN A 150.214.5.84
smtp2.cica.es. 172800 IN A 150.214.5.100
;; Query time: 83 msec
;; SERVER: 150.214.5.83#53(dns1.cica.es)
;; WHEN: Thu Jan 20 17:54:54 2005
;; MSG SIZE rcvd: 295
```

Los registros A y NS ya nos son conocidos. Aparece el registro SOA

```
cica.es. 172800 IN SOA dns1.cica.es. hostmaster.cica.es
```

que indica quién es la autoridad para el dominio cica.es.

También nos encontramos con registros MX, que a pesar de tener una gran importancia no son muy conocidos.

```
cica.es. 300 IN MX 10 smtp.cica.es.
cica.es. 300 IN MX 15 smtp2.cica.es.
```

Porqué dijimos que eran muy importantes, pues sencillamente porque dirigen los correos electrónicos. ¿Quién hoy día si le quitan el correo electrónico se quedaría igual?. Pues estos registros dicen que para todas las direcciones de correo electrónico del dominio cica.es, como por ejemplo jperez@cica.es, deben dirigirse a los "intercambiadores de correo". Como es algo muy crítico, se suelen poner varios con una preferencia y en caso de fallo de alguno, los correos van al siguiente. En este caso irían preferentemente a `smtp.cica.es` y en caso de fallo de éste a `smtp2.cica.es`.

Preguntemos por un registro CNAME. El registro CNAME se suele utilizar como un alias o pseudónimo de otra u otras máquinas. ¿Qué utilidad puede tener ésto? Por ejemplo, los servicios de Internet suelen prestarse en direcciones estandarizadas. Si queremos ver el Boletín Oficial del Estado y no sabemos con certeza la dirección, una de las primeras que probaremos si tenemos cierta experiencia con internet será www.boe.es. Nuestra máquina con el servidor web, no tiene porqué llamarse www y además nos permite cambiar rápidamente a otra máquina sin demasiados problemas en nuestra red. Veamos lo que hace el CICA.

```
[root@nuevo pedro]# dig CNAME www.cica.es
; <<>>DiG 9.2.3 <<>>CNAME www.cica.es
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13142
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 1
;; QUESTION SECTION:
;www.cica.es. IN CNAME
;; ANSWER SECTION:
www.cica.es. 1719 IN CNAME ataman.cica.es.
;; AUTHORITY SECTION:
cica.es. 2368 IN NS dns2.cica.es.
cica.es. 2368 IN NS chico.rediris.es.
cica.es. 2368 IN NS sun.rediris.es.
cica.es. 2368 IN NS dns1.cica.es.
;; ADDITIONAL SECTION:
dns1.cica.es. 152414 IN A 150.214.5.83
;; Query time: 2 msec
;; SERVER: 192.168.2.2#53(192.168.2.2)
;; WHEN: Thu Jan 20 18:00:37 2005
;; MSG SIZE rcvd: 150
```

La línea importante es la que nos dice que www.cica.es es un apodo (CNAME) de la máquina ataman.cica.es. Si esa máquina se cae, una posible solución es cambiar el registro CNAME de www.cica.es a atamon.cica.es, que es una máquina que tenemos preparada para ello. El resto de usuarios (de todo el mundo) seguirán apuntando sus navegadores a www.cica.es sin enterarse del problema.

El recurso PTR es un poco más complicado. Veamos. Para que el mismo sistema funcione tanto para pedir conversiones de direcciones simbólicas a direcciones IP, como al revés, de direcciones IP a direcciones simbólicas se crea el recurso PTR y un dominio especial de nombre in-addr.arpa.

Un comando sencillo para saber el nombre que le corresponde a una dirección IP es el comando host

```
[root@nuevo pedro]# host 150.214.5.119
119.5.214.150.in-addr.arpa domain name pointer ataman.cica.es.
```

Vemos que nos devuelve que se corresponde con la dirección simbólica ataman.cica.es, pero antes da una información un poco rara. Como en las direcciones simbólicas la jerarquía va de derecha a izquierda y en las direcciones IP de izquierda a derecha, se emplea un truco. Todas las direcciones IP se colocan bajo el dominio in-addr.arpa y se va poniendo cada uno de los bytes de la dirección IP de derecha a izquierda. Así `150.214.5.119` queda como `119.5.214.150.in-addr.arpa`. Veamos qué dice nuestro amigo dig sobre esto:

```
[root@nuevo pedro]# dig PTR 119.5.214.150.in-addr.arpa
; <<>>DiG 9.2.3 <<>>PTR 119.5.214.150.in-addr.arpa
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31031
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 13
;; QUESTION SECTION:
;119.5.214.150.in-addr.arpa. IN PTR
;; ANSWER SECTION:
119.5.214.150.in-addr.arpa. 3426 IN PTR ataman.cica.es.
;; AUTHORITY SECTION:
....
;; Query time: 4 msec
;; SERVER: 192.168.2.2#53(192.168.2.2)
;; WHEN: Thu Jan 20 18:07:41 2005
;; MSG SIZE rcvd: 491
```

Correcto, es un hacha este dig. Nos dice que estamos hablando de ataman.cica.es y es un registro de tipo PTR (Poin Te R).

5.4. Servidores de Nombres

Seguro que el DNS os ha deparado muchas sorpresas. Pues aún hay más. El hecho de configurar un Servidor de Nombres es una auténtica odisea.

El servidor de nombres por excelencia es el demonio named, que es parte del paquete BIND, preparado y coordinado por el Internet Software Consortium.

Un servidor de nombres puede estar configurado de alguna de estas formas:

master

Es el "dueño" del dominio[2], en el que se hacen las modificaciones para ese dominio, responde las consultas que se le hagan y se encarga de propagarlo al resto.

slave

Son servidores de nombres del dominio y así se encargan de resolver las preguntas que se les hagan. Pero cada cierto tiempo le preguntan al "master" del que dependen para actualizar su información.

caching-only

Solamente constituyen un caché de datos para optimizar las respuestas. Por ejemplo, podemos montar uno de este tipo en nuestro u organización para que todos los puestos clientes le pregunten a él. Sirve para optimizar las respuestas y el uso de la línea de comunicaciones, pero además simplifica la política de seguridad. Para las peticiones de resolución DNS, los clientes no pueden atravesar el cortafuegos y sí esta única máquina.

[2] Zona es el término empleado.

forwarding

Redirige las peticiones a otros servidores de nombres. Es poca la diferencia con el de caché.

En el terreno árido, BIND guarda su configuración los siguientes sitios:

[/etc/named.conf]Fichero de configuración del demonio named.

[/var/named/]Directorio en el que almacena el resto de ficheros.

Veremos el caso más completo que es el de una zona master. Crearemos el dominio midominio.org.

Una vez que hemos creado la zona para nuestro dominio, le añadiremos registros, que pueden ser de los tipos vistos anteriormente (A, CNAME, NS o mX).

Crearemos un registro de tipo A. La dirección simbólica servidor.midominio.org la asignamos a la dirección IP 192.168.12.2. Para la resolución inversa (PTR) tendremos que crear el dominio inverso 12.168.192.in-addr.arpa.

Añadiremos un registro CNAME y creamos un alias entre las direcciones simbólicas www.midominio.org y servidor.midominio.org.

Empezamos con /etc/named.conf

```
[root@nuevo pedro]# more /etc/named.conf
// generated by named-bootconf.pl
//
// a caching only nameserver config
//
options {
/*
* If there is a firewall between you and nameservers you want
* to talk to, you might need to uncomment the query-source
* directive below. Previous versions of BIND always asked
* questions using port 53, but BIND 8.1 uses an unprivileged
* port by default.
*/
// query-source address * port 53;
directory "/var/named";
};
controls {
inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." {
type hint;
file "named.ca";
};
zone "localhost" {
allow-update { none; };
type master;
file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" {
allow-update { none; };
type master;
file "named.local";
};
zone "midominio.org" {
type master;
file "midominio.org.zone";
};
include "/etc/rndc.key";
```

Especifica que los ficheros de zonas y configuración adicional estará en el directorio /var/named. La configuración que viene por defecto crea un servidor de nombres que funciona como caché. De ahí provienen las zonas ".", localhost y 0.0.127.in-addr.arpa.

Para la zona que hemos creado, midominio.org, especifica que es de tipo master y que el resto de la configuración se encuentra en el fichero midominio.org.zone, que se encontrará en el directorio /var/named. Veámoslo.

```
[root@nuevo pedro]# more /var/named/midominio.org.zone
$TTL 86400
servidor.midominio.org. IN SOA localhost root (
2004021207 ; serial
28800 ; refresh
14400 ; retry
3600000 ; expire
86400 ; ttl
)
servidor IN A 192.168.12.2
.midominio.org IN MX 1 servidor.midominio.org.
```

La autoridad para el dominio (SOA) es servidor.midominio.org, vemos que el serial es 2004021207. Normalmente, por convención se pone en formato año, mes, dia, modificación dentro del día. El resto de valores son el tiempo en segundos, en que se refresca la información a los esclavos, que se reintenta en caso de no poder conectar, tiempo de expiración y máximo tiempo que lo pueden tener las cachés.

Hemos creado un registro tipo A que une las direcciones servidor.midominio.org y 192.168.12.2 y también un registro MX que indica que el correo dirigido al dominio midominio.org, será recogido por el servidor servidor.midominio.org.

En el fichero named.local podemos observar una típica zona de registros inversos tipo PTR.

```
[root@nuevo pedro]# more /var/named/named.local
$TTL 86400
@ IN SOA localhost. root.localhost. (
1997022703 ; serial
28800 ; refresh
14400 ; retry
3600000 ; expire
86400 ; ttl
)
@ IN NS localhost
1 IN PTR localhost.
2 IN PTR servidor.midominio.org.
```