Ftp Anónimo COMO

Rodolfo García Peñas, `kix@mad.servicom.es` v1.0-Sayaka, 1 Septiembre 1998

Este COMO describe cómo instalar, configurar y mantener un servidor ftp de una manera fácil y rápida.

Índice General

1 Editor.

La idea inicial de escribir este documento ha sido sin duda el irc. Hay mucha gente que entra y dice que quiere poner un servidor ftp corriendo en su sistema y claro está que eso no se hace en 10 minutos.

1.1 Copyright y cosas de esas.

Este como ha sido escrito por *Rodolfo García* (anda!! si soy yo!!), pero simplemente para mantener una misma línea de trabajo me gustaría que si hay que hacer modificaciones de algo, se me indique y yo muy gustosamente lo haría. Esto permitirá que dentro de 2 años nos encontremos con 16 Cacho-A-Ftp-Anonimo-HowTo incompletos y se tenga que perder un tiempo para organizarlo. Por otro lado no me hago responsable de los posibles problemas, pérdidas, ... de datos o de seguridad por seguir este Como.

1.2 Dedicatorias ;-).

Pues si, este ftp va dedicado a todo el mundo del canal `#linux` de irc-hispano, a los que trabajan por Linux en España, en especial a *Ismael Olea*, *Paco Pepe* `Pukka` :-) (ehh y a su preciosa *Viki* :-P), a *Juan José Amor Iglesias* y a *Ramón* Gutiérrez/, `CamuS` :-).

A *Sayaka*, una gran amiga de la cual no creo pueda nunca olvidarme, por ello este como lleva su nombre en el principio (v1.0-Sayaka).

Por otro lado quiero saludar a mis compañeros y amigos de universidad (`http://www.upco.es`).

1.3 Colaboraciones.

En especial, quiero dar las gracias a Ignacio Arenaza, que me permitió hacer unos ajustes a este COMO y a Francisco José Montilla, el cual me corrigió dos errores muy gordos.

1.4 Realización.

Para editar este COMO, no he seguido nada, lo he escrito desde el principio con lo que sabía, con mi poca experiencia y mirando manuales y demás. Una vez escrito, lo he complementado, ajustado,... con la *FAQ* sobre ftp anónimo que ha escrito Christophe R. Klaus (`cklaus@iss.net`).

2 Requisitos.

1. Tener instalado Linux, con un kernel 2.0.x (al menos), pero no he probado con menos, se aceptan reclamaciones.

2. Tener un paquete de ftp apropiado, en mi caso instalaré `wu-ftp` en Debian, aunque lo explicaré para todas las distribuciones.

3 Instalación.

La instalación en las distintas distribuciones es muy variable, pero todas se basan en lo mismo. Por ello empezaré en Debian que es la que yo uso, y pasaré a RedHat y SlackWare hasta donde vea que puedo.

En Debian es muy fácil, buscamos donde tenemos el paquete, y ejecutamos lo siguiente:

```
dpkg -i "nombre de paquete"
```

En mi caso, sería:

```
dpkg -i wu-ftpd-academ_2.4.2.16-9.deb
```

Una vez realizado esto, se nos preguntarán unas cuestiones.

```
Do you want to set up or update an anonymous FTP account now? [n]
```

Esta pregunta nos indica si queremos crear una cuenta de ftp anónimo en este momento. Nosotros indicaremos que sí, puesto que queremos una configuración de este tipo, pero si no la deseamos decimos que no y no ocurre nada más.

Al decir que sí, nos pregunta de nuevo:

```
Enter the name of the FTP home directory: [/home/ftp]
```

Debemos indicar dónde situaremos el directorio por defecto del ftp, por defecto `/home/ftp` , donde nosotros lo dejaremos.

```
/home/ftp does already exist, should I use it? [n]
```

Diremos que sí para crear el directorio raíz del ftp. La siguiente pregunta, será si deseamos crear un directorio para permitir que nos suban cosas, nosotros diremos que sí, aunque luego se explicará cómo configurarlo para activar o desactivar esta cuestión.

Con todo esto, tendríamos instalado el ftp, pero queda configurarlo.

Para comprobar que está correcto, realizaremos lo siguiente:

```
<kix@hell>$ ftp 127.0.0.1
Connected to 127.0.0.1.
220-Welcome, archive user [unknown]@localhost !
220-
220-The local time is: Mon Jul 27 22:04:24 1998
220-
220-This is an experimental FTP server.  If have any unusual problems,
220-please report them via e-mail to <root@hell.darkness.org>.
220-
220-If you do have problems, please try using a dash (-) as the first
220-character of your password -- this will turn off the continuation
220-messages that may be confusing your ftp client.
220-
220 hell.darkness.org FTP server (Version wu-2.4.2-academ[BETA-16](1) Sat
May 23 19:56:02 CEST 1998) ready.
Name (127.0.0.1:root):
```

Si está usando RedHat, Slackware o cualquier otra distribución, no se preocupe, posteriormente se indicará cómo configurar todo a mano. Aun así, indicare como se instala:

```
rpm -i "nombre_del_paquete_ftpd.rpm"
rpm -i "nombre_del_paquete_anon-ftp.rpm"
```

Esto es debido a que son dos paquetes, el del ftp anónimo, que en RedHat 5 se llama `anonftp-2.5-1.i386.rpm` y el del demonio avanzado de ftp, para nosotros el wu-ftpd, `wu-ftpd-2.4.2b16-4.i386.rpm`.

SlackWare:

```
tar -xvzf "nombre_del_paquete.tgz"
```

4 Configuración.

La configuración de un ftp es cosa fácil, pero debemos estar atentos por la seguridad del sistema.

4.1 Cuenta de usuario.

Lo primero será añadir el usuario `ftp`, que en Debian se ha creado; pero deberíamos repasar como está, en RedHat también aparece, pero en SlackWare no se crea. Para ello editaremos el archivo `/etc/passwd` y miraremos si tenemos una línea parecida a esta:

```
ftp:*:100:101::/home/ftp:/bin/false
```

Si no fuera así, la crearíamos.

1. `ftp`: sería el nombre de usuario. (Fíjese que siempre es `ftp`, sin distinción del nombre del servidor de ftp — wu-ftpd en nuestro caso —).
2. `*`: Anulamos la capacidad de acceso shell al sistema del usuario `ftp`. Como el demonio ftp necesita una, le pondremos una falsa, de este modo el acceso al sistema como usuario ftp queda estrictamente limitada a la función de ejecutar el demonio.
3. `100`: UID de usuario, puede variar.
4. `101`: Grupo del usuario, que es variable también.
5. `/home/ftp`: Este es el directorio raíz del ftp para el usuario `anonymous`. Si en este campo apareciera "/" compartiríamos todo nuestro linux, **muy inseguro**. En el caso de usuarios del sistema, el directorio `home` sería el del usuario en cuestión.
6. `/bin/false`: Este será el nombre del shell a ejecutar. Como queremos limitar el acceso shell del usuario `ftp`, pero el demonio ftp necesita una shell que *exista* ponemos `/bin/false`, asegurándonos de que exista. Si no es así, la creamos:

   ```
   #!/bin/sh
   exit 1
   ```

   Los permisos recomendables para este fichero son `rwxr-xr-x`.

   Además, esta *shell* ha de aparecer listada en el fichero `/etc/shells`, por lo que la habremos de añadir si no lo está.

4.2 /etc/ftpusers

Lo segundo es restringir los nombres de usuario que nunca deben entrar al ftp. Para ello editaremos el archivo `/etc/ftpusers` e introduciremos los que queramos:

```
# /etc/ftpusers: list of users disallowed ftp access. See ftpusers(5).

root
daemon
bin
sys
.....
```

Con esto cuando un usuario incluido en este archivo intente entrar, el acceso le será denegado.

Si mantenemos un servidor para distintos usuarios, cuando se introduzca un nombre y un password válido, el usuario aparecerá en su directorio `$HOME`, y podrá acceder a aquellas partes del sistema a las que tendría acceso por shell.

Si no queremos que acceda al servidor por ftp, añadiremos su nombre de usuario a este archivo.

5 La shell.

El shell de usuario debe aparecer en `/etc/shells` pero si no aparece no ocurre nada y además deberá tener en cuenta que la restricción de no ponerlo no aumenta la seguridad.

6 El directorio del ftp.

El directorio `/home/ftp` en nuestro caso debe cumplir lo siguiente, para que sea seguro:

6.1 /home/ftp

El directorio raíz del ftp (`/home/ftp`) debe ser propiedad del usuario root y con acceso de escritura prohibido al resto de usuarios, pero sí de ejecución para todos (`chmod 555`)

Como ejemplo:

```
dr-xr-xr-x   6 root     root          1024 Jul 19 21:16 ftp
```

Es importante saber que algunas revistas, y manuales sobre ftp recomiendan que el usuario propietario de `/home/ftp` sea ftp, pero esto es un error enorme si queremos tener un servidor seguro.

6.2 /home/ftp/bin

El directorio `/home/ftp/bin` debe ser propiedad del root y sin acceso de escritura al resto de los usuarios. Como ejemplo:

```
d--x--x--x   2 root     root         1024 Jul 27 21:48 bin
```

Dentro de este directorio, los archivos que tendremos serán todos los ejecutables, en mi caso son:

```
---x--x--x   1 root     root        15204 Oct 24  1997 compress
---x--x--x   1 root     root        46324 Nov  6  1997 cpio
---x--x--x   1 root     root        45660 Jul 27 21:48 gzip
---x--x--x   1 root     root        29404 Jul 27 21:48 ls
---x--x--x   1 root     root        62692 Oct 20  1997 sh
---x--x--x   1 root     root       103968 Jul 27 21:48 tar
```

Como se puede ver todos ellos son pertenecientes a `root` y con permisos de sólo ejecución para todo el mundo (modo `111`). Los comandos de este directorio son los mismos que usamos en nuestro sistema, por ejemplo "`ls`", los copiamos aquí si no los tenemos y les aplicamos los permisos.

6.3 /home/ftp/etc

El directorio `/home/ftp/etc` tendrá permiso de solo ejecución y será propiedad de `root`:

```
d--x--x--x   2 root     root         1024 Jul 27 23:42 etc
```

Los archivos siguientes son los que tendremos en este directorio:

```
-r--r--r--   1 root     root           18 Jul 27 21:48 group
-r--r--r--   1 root     root         8376 Nov  6  1997 ld.so.cache
-r--r--r--   1 root     root           44 Jul 27 21:48 passwd
-r--r--r--   1 root     root          172 Jul 19 21:16 pathmsg
```

Todos los archivos deben ser de sólo lectura (modo `111`).

Tanto el archivo `group` como el archivo `passwd` de este directorio deben contener la información mínima posible. Un ejemplo claro sería:

`passwd`:

```
root:*:0:0:root::
ftp:*:10::Anonymous FTP::
```

`group`:

```
root::0:
staff:::
```

El archivo `pathmsg` es un archivo de información para el usuario cuando introduce algún carácter no valido. Es editable por nosotros para que ponga lo que queramos.

`pathmsg` Original:

```
You used an illegal filename. The filenames for anonymous
users must be made of only the characters  A-Z,  a-z, 0-9
and "._-+" and may not begin with a "." or a "-".
```

`pathmsg` mío actual:

```
Baby estas usando un nombre d'archivo erroneo. A ver zi te
enteras ke solo puedes usar los caracteres A-Z, a-z, 0-9
y "._-+" pero sin empezar por "." o "-".
```

Fuera de las coñas y demás, es "algo" más inteligible que en inglés.

El archivo `ld.so.cache` contiene los nombres de las librerías necesarias, recomiendo no tocarlo si no es necesario.

6.4 /home/ftp/lib

El directorio `/home/ftp/lib` donde las librerías son contenidas, debe tener propiedades de solo ejecución (`d--x--x--x`). En el interior encontraremos librerías y links a las mismas. Debemos tener en cuenta que deben tener permisos de sólo lectura, pero que los links tienen todos los permisos activados.

Como ejemplo:

```
-r--r--r--   1 root     root       164803 Jul 27 21:48 ld-2.0.5.so
lrwxrwxrwx   1 root     root           11 Jul 14 02:23 ld-linux.so.2 -> ld-2.0.5.so
```

6.5 /home/ftp/pub

El directorio `/home/ftp/pub` es donde se alojarán los archivos que queremos compartir. Su propietario debería ser `ftp`, y los accesos serán de ejecución y lectura:

```
dr-xr-xr-x  21 ftp      ftp          1024 Jul 26 20:04 pub
```

El permitir la escritura en este directorio no es una buena idea, puesto que los archivos deberían estar controlados, para ello usaremos un directorio concreto para las subidas. La recomendación es tener los permisos de todo lo que cuelga de este directorio (`pub`) como `555`.

En otros sistemas como SunOS las propiedades de este directorio son `2555`, de tal forma que se pueden crear nuevos archivos con el mismo grupo.

6.5.1 /home/ftp/pub/incoming

El directorio `/home/ftp/pub/incoming` es el directorio donde permitiremos la subida de ficheros a nuestra máquina. Por ello debe tener las siguientes propiedades:

```
drwxr-x-wx   2 root     root         1024 Jul 19 21:16 incoming
```

Como se puede observar tiene permiso de ejecución (como es un directorio, realmente es permiso de *acceso* al directorio) y de escritura, pero no de lectura. Esto se hace para que sólo personas autorizadas vean el contenido de este directorio. Si quisiéramos que fuera un directorio para compartir ficheros, solo tendríamos que añadir la propiedad de lectura.

Por otro lado, tendremos que editar un archivo de configuración para que todo esté perfecto. Este archivo se verá en la parte de configuración avanzada.

7 El arranque.

En este momento el demonio de ftp estaría preparado para ser lanzado, los directorios con los permisos adecuados y todo a punto para arrancar el demonio. Quedan dos partes importantes, la forma de arrancarlo y los archivos a ofrecer.

Para el arranque del demonio, lo mejor es dejar que `inetd` lo lance cuando sea necesario, para ello se lo especificamos en el archivo `/etc/inetd.conf`, en el cual tendremos que poner (si no existe) una línea como esta:

```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd /usr/sbin/ftpd
```

En algunos sistemas en lugar de `fptd` puede ser `in.ftpd`.

Ahora podemos lanzar un ftp sin archivos, pero funcional reiniciando el `inetd` con el comando "`killall -HUP inetd`"

8 Los archivos a ofrecer.

Los archivos a ofrecer los introduciremos en el directorio `pub` de nuestro ftp, pudiendo crear directorios, y teniendo en cuenta los permisos para que no sean modificables, borrables y demás. Un ejemplo sería:

```
-r--r--r--   1 root     root       640005 Jun 23 23:31 WindowMaker.gz
dr-xr-xr-x   2 root     root         2048 Jul 20 21:06 apps
dr-xr-xr-x   2 root     root         1024 Jul 20 21:06 themes
```

9 Configuración avanzada.

Hasta este punto tendríamos un ftp seguro y totalmente funcional, pero el ftp nos permite muchas mas cosas, me referiré al del *wu-ftpd* en especial, los demás servidores son iguales o muy parecidos.

Los archivos se alojan en `/etc/` o en `/etc/directorio` en el caso de `wu-ftp-academ` es `/etc/wu-ftpd-academ`. Para buscar los suyos lo mejor es:

```
find /etc -name *ftp*
```

9.1 El archivo `ftpaccess`.

Este archivo es el que tiene toda la configuración del ftp, por lo cual si montamos el ftp desde cero tendremos que repasarlo a tope.

9.1.1 Mensajes.

Los mensajes son un punto importante de información. Podemos encontrarlos de tres tipos, pero todos se configuran en este archivo.

El primero es el que recibimos nada más conectar con el servidor de ftp, que suele ser de tipo informativo. Es indicado por una línea como esta en el archivo `ftpaccess`:

```
banner /etc/wu-ftpd-academ/welcome.msg
```

Teniendo en cuenta que el fichero `/etc/wu-ftpd-academ/welcome.msg` tiene un contenido deseado para nosotros.

Otro tipo de mensaje es el que aparece nada más autorizarnos con el servidor, y va indicado por esta línea:

```
message /welcome.msg           login
```

Teniendo en cuenta que la `/` indica el directorio raíz del ftp, en nuestro caso `/home/ftp`, puesto que hemos accedido ya.

El otro tipo de mensajes es el que aparece al cambiar de directorio y que normalmente nos informa de lo que allí tenemos. Viene indicado por la línea siguiente:

```
message .message               cwd=*
```

Como se puede ver, es un archivo llamado `.message` que aparece en cada cambio de directorio, y se muestra con el cwd. (`cwd=*`).

Podemos cambiar lo que queramos y añadir un tipo de mensaje para otras acciones deseadas.

9.1.2 Información del servidor de ftp.

Es importante que los que accedan a nuestro servidor puedan indicarnos cualquier error. Para ello, debemos dejar una dirección de correo electrónico bajo una línea similar a esta:

```
# direccion email del administrador ftp,
# puede utilizarse con la macro %E
# en los ficheros de mensajes.
        email usuario@servidor_correo
```

10 Accesos.

Podemos configurar los accesos: en primer lugar, es posible especificar un número máximo de intentos de conexión (logins) fallidos permitidos por el servidor como seguridad, aunque en el caso del ftp anónimo no sería fundamental. Para ello escribiremos la línea siguiente en el archivo `ftpaccess`:

```
loginfails X
```

Donde X es el número de intentos.

Por otro lado podemos restringir el dominio de acceso, que puede ser local a la máquina, remoto a la máquina o los dos. Se indica bajo unas líneas de este tipo:

```
class  local   real,guest,anonymous *.my.domain 192.168.0.0
class  remote  real,guest,anonymous *
class   all    real,guest,anonymous *
```

1. La primera permite un acceso local para los usuarios reales, invitados y anónimos para el dominio indicado, en este caso `*.my.domain 192.168.0.0`.
2. La segunda permite un acceso remoto para toda la red (*) para los usuarios reales de la máquina servidora, los invitados y el anónimo.
3. La tercera permite un acceso total para los mismos usuarios y para toda la red.

Podemos restringir también el número máximo de usuarios que pueden acceder simultáneamente. Esto es necesario para no sobrecargar la máquina, la línea telefónica, la red,... Tenemos una restricción para cada tipo de acceso, un ejemplo sería:

```
limit  local   20     Any   /etc/wu-ftpd-academ/msg.toomany
limit  remote  100    Any   /etc/wu-ftpd-academ/msg.toomany
limit   all    10     Any   /etc/wu-ftpd-academ/msg.toomany
```

Así limitamos el acceso local a 20 usuarios, el remoto a 100 y el total a 10, hay que tener en cuenta que las líneas local y remota estaban comentadas. Si por algún motivo intenta entrar un usuario más al sistema y sobrepasa este valor, se le mostrará el mensaje escrito en nuestro caso en `/etc/wu-ftpd-academ/msg.toomany`. Como el lector supondrá, todo esto es modificable.

10.1 Modificación de archivos.

Un tema importante es permitir o no la modificación de las propiedades de los archivos. Por ello el archivo `ftpaccess` dispone de unas líneas para ello:

```
#accion       yes/no     usuario
#--------------------------------
rename          no      anonymous    # rename permission?
delete          no      anonymous    # delete permission?
overwrite       no      anonymous    # overwrite permission?
chmod           no      anonymous    # chmod permission?
umask           no      anonymous    # umask permission?
```

En este caso, todos los permisos para el usuario anonymous están desactivadas.

10.2 Archivos no bajables.

Existirán archivos que no deseamos que sean obtenibles, como pudieran ser `/etc/passwd` y `/etc/group`, del directorio del ftp (en nuestro caso (`/home/ftp/etc/passwd` y `/home/ftp/etc/group`). Para ello tendríamos la siguiente línea en el archivo `ftpaccess`:

```
# estos ficheros no son bajables
noretrieve /etc/passwd /etc/group
noretrieve core
```

La primera línea es un comentario, la segunda hace lo explicado anteriormente, mientras que la tercera no permite la bajada de archivos `core` del ftp. Esto es debido a que los `core` son volcados de memoria y pueden contener información valiosa.

10.3 Subidas al servidor.

Antes se ha indicado como preparar un directorio para las subidas al servidor de ftp (directorio `incoming`). Pero en este archivo vamos a poner unas líneas muy especiales:

```
upload  /home/ftp   *              no
upload  /home/ftp   /pub/incoming  yes     ftp     daemon  0666 nodirs
```

La primera línea indica que las subidas a `/home/ftp`, en cualquiera de sus directorios, no están permitidas.

La segunda indica específicamente que las subidas a `/home/ftp/pub/incoming` están permitidas, pero no se pueden crear directorios y el permiso de los ficheros será `0666` (modo `0666`).

El directorio `incoming` puede ser un problema muy grande, en muchos sitios se recomienda un sistema de archivos propio, puesto que si se llena el del sistema por exceso de *uploads* de ftp, tendríamos grandes problemas.

Con todo esto, todo el ftp estará perfectamente configurado, pero aun así, existen más opciones, como indicar alias escribiendo :

```
        alias   incoming:       /pub/incoming
```

El usuario al poner `cd incoming` pasara automáticamente a `/pub/incoming`, pero básicamente este archivo ya estaría perfecto.

10.4 El archivo `ftpconversions`.

Este archivo es muy útil para bajar archivos que no están comprimidos o empaquetados. En casos como el *wu-ftpd* es muy conveniente poner comandos como `gzip`, `gnutar`, `compress`,... compilados y enlazados de forma estática, en el directorio `/home/ftp/bin` con los permisos correspondientes (`111`) para permitir bajadas de archivos a la vez que los comprimimos. Su aspecto es el siguiente:

```
:.Z:  :  :/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
:   : :.Z:/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
:.gz: :  :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
:   : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
:   : :.tar:/bin/tar -c -h -f - %s:T_REG|T_DIR:O_TAR:TAR
:   : :.tar.Z:/bin/tar -c -h -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRE
:   : :.tar.gz:/bin/tar -c -h -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
:   : :.tgz:/bin/tar -c -h -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
:   : :.ltar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
:   : :.ltar.Z:/bin/tar -c -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
:   : :.ltar.gz:/bin/tar -c -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
```

Ejemplo:

```
-r--r--r--   1 root     root       795509 Jan 20  1998 tel2308a
-r--r--r--   1 root     root       797901 Jan 20  1998 tel2308b
```

Son los ficheros a bajar del servidor. Como se ve, no están en forma comprimida ni empaquetada, pero queremos bajarlo como `tar.Z`. Esto a priori se presenta como un problema, puesto que son dos comandos en uno, por ello podemos usar un programa que nos permita pipes (`|`) en nuestro ftpd. Con el siguiente, copiado al directorio `bin` de nuestro ftp, después de compilarlo, el problema esta casi resuelto:

```
/* pipe.c: ejecuta dos comandos en una pipe */
/* codigo escrito por Gary Mills */

#define NULL (char *)0
#define MAXA 16

main(argc, argv) int argc; char *argv[]; {
char *av1[MAXA], *av2[MAXA];
int i, n, p[2], cpid;

i = 0; n = 0;
while ( ++i < argc && n < MAXA ) {
if ( *argv[i] == '|' && *(argv[i]+1) == '\0' ) break;
av1[n++] = argv[i];
}
if ( n == 0 ) uexit();
av1[n] = NULL;
n = 0;
while ( ++i < argc && n < MAXA )
av2[n++] = argv[i];
if ( n == 0 ) uexit();
av2[n] = NULL;
if ( pipe(p) != 0 ) exit(1);
if ( ( cpid = fork() ) == (-1) ) exit(1);
else if ( cpid == 0 ) {
```

```
(void)close(p[0]);
(void)close(1);
(void)dup(p[1]);
(void)close(p[1]);
(void)execv(av1[0], av1);
_exit(127);
}
else {
(void)close(p[1]);
(void)close(0);
(void)dup(p[0]);
(void)close(p[0]);
(void)execv(av2[0], av2);
_exit(127);
}
/*NOTREACHED*/
}
uexit() {
(void)write(2, "Modo de uso: pipe  | \n", 34);
exit(1);
}
```

El programa lo tenemos, el problema es indicarle al ftpd que lo use. Para ello pondremos esta línea en el archivo `ftpconversions`:

```
:  :  :.tar.Z:/bin/pipe /bin/tar cf - %s | /bin/compress -c:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
```

ATENCION: La línea puede estar partida !!! Espero que se entienda, si hay dudas, por favor indíquemelas.

De esta forma, el comando para obtener un `.tar.Z` es llamando a `bin/pipe` que a su vez llama a `bin/tar` y a `bin/compress` (con sus opciones correspondientes). Este comando se puede ampliar para otros comandos como pueden ser `gzip` o `zip` junto con el `tar`.

La variedad solo tiene límite con las combinaciones de compresores.

11 Puntos clave a recordar.

Si tiene un agujero de seguridad en su ftpd lo mejor es que lo cierre temporalmente, obtenga el último demonio de ftp sin errores o con errores parcheables, y sustituya el demonio anterior. Asegúrese de que los archivos tienen los permisos correspondientes, que el archivo `/etc/passwd` de su ftp es correcto, y ponga el seguimiento de su ftp de forma muy alta y que haga logs de las conexiones. Si aun así tiene problemas, simplemente rece.

El ftp anónimo no es más que un servicio ofrecido a personas externas a nuestro sistema, por ello se crea el directorio `/home/ftp`, pero se puede tener un ftp para los usuarios del sistema pero no para los externos, por lo cual no es un servidor anónimo de ftp.

Recuerde que el directorio home será por tanto el del usuario que acceda por ftp. Siguiendo este COMO le puede ayudar a configurarlo sea cual sea su caso.

Si desea más seguridad debería compilar todo usted mismo, de forma estática.

12 Archie.

El servicio Archie es un simple buscador de archivos sobre máquinas con ftp. Podemos buscar un archivo que necesitamos usando este servicio sobre cientos de máquinas de todo el mundo de una forma rápida y efectiva al 100%. Por otro lado podemos incluir nuestro servidor en un buscador Archie (sólo si se esta siempre conectado, es un poco absurdo para un acceso temporal de 30 minutos diarios), para ello debemos mandar un mail a la lista de buscadores Archie (`archie-updates@bunyip.com`) para los siguientes servidores Archie:

```
archie.ac.il              132.65.20.254    (Israel)
archie.ans.net            147.225.1.10     (ANS, NY (USA))
archie.au                 139.130.4.6      (Australia)
archie.doc.ic.ac.uk       146.169.11.3     (United Kingdom)
archie.edvz.uni-linz.ac.at 140.78.3.8      (Austria)
archie.funet.fi           128.214.6.102    (Finlandia)
archie.internic.net       198.49.45.10     (ATT, NY (USA))
archie.kr                 128.134.1.1      (Korea)
archie.kuis.kyoto-u.ac.jp 130.54.20.1      (Japon)
archie.luth.se            130.240.18.4     (Swedish)
archie.ncu.edu.tw         140.115.19.24    (Taiwan)
archie.nz                 130.195.9.4      (Nueva Zelanda)
archie.rediris.es         130.206.1.2      (Spain) (Este es el nuestro)
archie.rutgers.edu        128.6.18.15      (Rutgers University (USA))
archie.sogang.ac.kr       163.239.1.11     (Korea)
archie.sura.net           128.167.254.195  (SURAnet (USA))
archie.sura.net(1526)     128.167.254.195  (SURAnet alt. MD (USA))
archie.switch.ch          130.59.1.40      (Swiss Server)
archie.th-darmstadt.de    130.83.22.60     (Alemania)
archie.unipi.it           131.114.21.10    (Italia)
archie.univie.ac.at       131.130.1.23     (Austria)
archie.unl.edu            129.93.1.14      (U. of Nebraska, Lincoln (USA))
archie.univ-rennes1.fr                     (Francia)
archie.uqam.ca            132.208.250.10   (Canada)
archie.wide.ad.jp         133.4.3.6        (Japon)
```

13 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: `www.insflug.org`. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Francisco José Montilla, `pacopepe@insflug.org`.