MIRROR

AUTOR: Juan Carlos Martín Severiano

1.- OBJETIVOS Y DESCRIPCIÓN DEL SERVICIO MIRROR

*OBJETIVOS

Mirror es una herramienta escrita en PERL diseñada para duplicar una jerarquía de directorios entre dos máquinas, normalmente remotas. Como su propio nombre indica da lugar a réplicas exactas de lo que contiene un determinado servidor, permitiendo la aparición de servidores alternativos para la descarga de ficheros. Evidentemente los candidatos a ser duplicados son máquinas de elevado número de accesos cuya distancia geográfica (o lentitud de transferencia inherente al tráfico que soporta) aconseja el tener un servidor más cercano para facilitar la descarga.

Crear un **mirror** de un servidor se asemeja por lo tanto a crear varias copias de un documento, en principio único en el mundo, y distribuirlas geográficamente del modo más adecuado para que quien esté interesado en acceder a sus contenidos pueda hacerlo de la manera más rápida y cómoda posible. En realidad quien hace la copia no tiene por qué tener nada que ver con el que hizo el original, y de hecho esto es normalmente lo que ocurre.

Este servicio puede ser muy interesante para redes en las que el acceso al exterior está muy limitado y se prefiere intercambiar recursos de espacio en disco duro por recursos de mayor ancho de banda para acceso al exterior. Si algún servidor exterior está muy demandado se generará mucho tráfico hacia fuera que podría ser ahorrado disponiendo del contenido que ofrece ese servidor externo, utilizando para ello la herramienta **mirror**.

Mirror utiliza FTP como protocolo para la transferencia del *site* original por lo que no sólo pueden utilizarlo administradores para hacer un duplicado en su máquina sino que además puede ser útil para cualquiera que quiera transferir un número elevado de ficheros mediante FTP. Es una característica muy interesante de la que adolecen muchos programas de FTP por lo que no debería olvidarse esta herramienta para la descarga masiva de ficheros. Otra característica muy especial que posee **mirror** es la de evitar la copia innecesaria de ficheros, lo cual agiliza sin duda el proceso de actualización. Para ello compara tanto el tamaño como la fecha de los ficheros y sólo transfiere aquellos que han sido modificados, o que no existían previamente.

Aunque originariamente **mirror** sólo estaba disponible para sistemas UNIX, actualmente está soportado por plataformas tipo WINDOWS 95/98 o NT.

En la fecha de la redacción de este documento, la versión de **mirror** era la 2.9. La última versión puede encontrarse en la siguiente dirección, empaquetada de dos formas diferentes, bien como ZIP o bien como TAR comprimido:

ftp://sunsite.org.uk/packages/mirror/mirror.tar.gz
ftp://sunsite.org.uk/packages/mirror/mirror.zip

Existe junto con **mirror** otra herramienta, mm, que se sirve de ésta última para manejar la copia de múltiples **mirrors** a la vez. Se encarga de la copia de tantos servidores como desee el usuario, empezando automáticamente la transferencia de uno cuando finaliza la anterior. Se ocupa de que sólo un **mirror** se esté duplicando en un determinado momento y permite la configuración de un tiempo mínimo entre intentos de duplicación de un determinado *site*.

*DESCRIPCIÓN

Mirror puede ser ejecutado de las siguientes dos maneras:

mirror [flags] -gsite:pathname

mirror [flags] [package-files]

El primer método se utiliza para transferir un fichero o un directorio remoto al directorio actual. Se recomienda que si es un directorio lo que se desea transferir que éste acabe en la barra normal con la que finalizan los directorios en UNIX ('/').

En el segundo método se requieren un mínimo número de argumentos y **mirror** es controlado por las líneas "campo=valor" (*keyword=value*) de los ficheros incluidos en el directorio "packages" que contienen la información acerca de la localización de los directorios remotos y cualquiera otra necesaria, como directorio destino, por ejemplo. Más adelante (en el punto 4.-Configuración) se verá la estructura concreta de estos ficheros. En caso de existir un fichero llamado "mirror.defaults", bien en el directorio que contenga al ejecutable **mirror** o bien en cualquiera incluido en el *path* definido por PERLLIB, entonces es cargado antes de cualquiera de los ficheros incluidos en "packages". Este fichero contiene las configuraciones por defecto que se utilizarán en todo proceso de duplicación si no se redefine su comportamiento en los "package-files" o ficheros del directorio "packages". Si no existe el fichero "mirror.defaults" se aplicarían las configuraciones por defecto que contiene el ejecutable **mirror**.

El proceso que sigue **mirror** puede dividirse en los siguiente pasos:

- Lo primero que hace es leer por turno los "package-files". Si el programa está actualmente conectado a un sitio distinto del que le corresponde duplicar lo que hace es desconectarse de ese sitio, cambiar al directorio local correspondiente (creándolo si es necesario) y acto seguido examina el directorio para obtener los detalles de todos y cada uno de los ficheros (y/o directorios) locales que estén en él. **Mirror** entonces trata de conectarse al sitio remoto contactando con el demonio ("daemon") de FTP que esté ejecutándose allí. Una vez conectado hace un *login* con la información contenida en los campos "remote_user" y "remote_password", y después examina el directorio remoto citado en el campo "remote_dir" mediante la orden FTP LIST, pasando el parámetro recursivo o no recursivo en función del valor del campo "recursive". También alternativamente, en caso de existir, puede ser examinado el fichero que contiene la información del listado del directorio remoto (habitualmente ls_lR_file o local_ls_lR_file).

- Una vez hecho esto se compara la información de los ficheros locales con la obtenida del sitio remoto creándose la lista de todos los ficheros remotos requeridos y el directorio local en el que se ha de almacenar cada uno de ellos.

- Los directorios destino necesarios se crean, y después se copian los ficheros desde el servidor remoto a sus respectivos directorios, normalmente en modo binario. Esto se hace almacenando cada fichero en un fichero temporal en el propio directorio local, fichero que puede ser comprimido o dividido en partes si es necesario. La fecha de este fichero se pone a la misma que tenía el fichero remoto. Finalmente se pone el nombre correcto al fichero temporal.

- Una vez que todos los ficheros han sido transferidos se crean todos los enlaces simbólicos (simbolic links) requeridos (en caso de que el sistema operativo los soporte). Hay que señalar que **mirror** no es capaz de manejar enlaces fuertes (hard links).

- Tampoco duplica información que no tenga sentido fuera del origen, como por ejemplo el propietario de los ficheros, ya sea en forma de usuario (owner) o de grupo (group). En caso de necesitar esta información se recomienda usar RDIST en su lugar.

A menos que se detecte un fallo interno al programa, cualquier error provocará el abandono del servidor actual para pasar al siguiente de la lista.

2.- REQUISITOS

Actualmente **mirror** está disponible tanto para sistemas que utilizan UNIX como para sistemas que utilizan WINDOWS 95/98/2000 o NT. La potencia requerida es mínima puesto que el programa se limita a la supervisión de la duplicación de ficheros a través de una red.

El único programa que **mirror** requiere tener previamente instalado es PERL versión 4, aunque es recomendable tener la versión 5 al menos.

Evidentemente también es necesaria una conexión a Internet o a la red de la que se pretenda tener el *espejo*, con un ancho de banda medianamente decente para hacer más rápido todo el proceso.

3.- ARQUITECTURA DEL SERVICIO

De la arquitectura que presenta el servicio **mirror** podría decirse que presenta un cliente que desea acceder a un servicio, un servidor que ofrece el servicio que pretende el cliente y en medio de ellos dos un segundo servidor que ofrece el servicio demandado de una manera más rápida y cercana al cliente.

4.- CONFIGURACIÓN

Primero describiremos el proceso de instalación del servicio y después como configurarlo para una correcta utilización.

Para llevar a cabo la instalación de **mirror** no es necesario contar con los privilegios de *root*. Dicho esto habría que hacerse con la última versión de **mirror** (disponible en las *url*

ftp://sunsite.org.uk/packages/mirror/mirror.tar.gz ó
ftp://sunsite.org.uk/packages/mirror/mirror.zip)

y disponerse a instalarla. Para ello hay que seguir los siguientes pasos:

- Primero hay que ejecutar la orden

  tar zxvf mirror.tar.gz o bien
  unzip mirror.zip

  en función del fichero que hayamos descargado.

- Comprobar si PERL está instalado en la máquina en la que vamos a instalar y configurar **mirror**. Si no lo está se puede visitar la dirección http://www.perl.com/ y obtener una copia de la última versión de PERL e instalarla.

- Ejecutar la orden

  perl install.ph here

  si se va a instalar **mirror** sin ser *root*. Para instalar **mirror** siendo *root* en el lugar que le corresponde (normalmente /usr/local/sbin) puede utilizarse el fichero *makefile* con la orden

  make install

  que instalará además las páginas del manual y las librerías perl necesarias (por si no las hubiera), además de configurar apropiadamente los permisos de ejecución.

Instalar **mirror** para cualquiera que tenga posibilidad de usar la máquina es un poco más complicado. En cualquier caso está contemplado en la documentación y la dejamos como referencia para todo aquel que quiera instalarlo de esta manera.

*OPCIONES

Aunque **mirror** tiene un gran número de parámetros que se le pueden pasar la mayoría de ellos sólo deberían ser usados para hacer un **mirror** muy simple como una ocasión aislada. Si la intención es mantener un servidor espejo habitualmente lo mejor es poner todos los detalles de la configuración en los "package-files" y ejecutar **mirror** sobre esos ficheros.

Las únicas opciones que deberían usarse de modo frecuente son –n y, en caso de que se quiera ver lo que **mirror** haciendo, -d

-d Activa depuración. Si este argumento se pone más de una vez (por ejemplo –d –d) el nivel de depuración se incrementa.

-n No hace nada, salvo comparar los directorios locales y remotos. No hay por lo tanto transferencia de ficheros.
El modo de depuración se pone a 2, así que se muestran trazas de lo que ocurriría si se ejecutara **mirror** sin esta opción.

-g site:path
Transfiere todos los ficheros incluidos en *path* y todos los subdirectorios incluidos en él. Si *path* no acaba en / se copiarán todos los contenidos del directorio que contenía al fichero incluido en *path.* Si lo que se quiere es copiar el contenido de un directorio ese directorio ha de acabar en /
(ejemplo host:/d1/directorioacopiar/).

-p package
Solo se copia el *package* indicado, incluso cuando se estén usando varios “package files”. Esta opción se puede poner todas las veces que se quiera copiándose todos y cada uno de los *packages* solicitados. Sin esta opción se hace un **mirror** de todos los “package files” que existan.

-R package
Similar a –p pero se salta todos los “package files” hasta que llega al *package* indicado. Útil para recomenzar un **mirror** que falló desde el lugar en que lo hizo.

-F Usa ficheros temporales dbm para recoger información sobre ficheros. Útil si se copia un directorio muy grande.

-r Equivalente a –k recursive=false.

-v Imprime los detalles de la versión de mirror.

-T No transfiere ningún fichero, solo cambia la fecha de los ficheros locales a la misma que tengan los ficheros remotos.

-Ufilename

Crea una lista de todos los ficheros transferidos por **mirror** y la almacena en el fichero dado por *filename*. Recordar que **mirror** cambia automáticamente al directorio indicado en "local_dir" para hacer su trabajo, así que habrá que poner el *path* completo del fichero *filename,* en caso de no estar en el directorio por defecto.
Si no se da el nombre de ningún fichero, se utiliza

upload_log.day.month.year

-k key=value

Sustituye el par key/value por el que aparezca en los ficheros de configuración. Ver Package Files más abajo.

-m Equivalente a –k mode_copy=true

-t Equivalente a -k text_mode=true

-f Equivalente a -k force=true

-s site Equivalente a -k site=site

-u user Equivalente a -k remote_user=user. Inmediatamente después se pide un password, que se utilizará como "remote_password".

-L Genera una versión impresa de la entrada y la salida.

*Package Files

Estos ficheros están divididos en una serie de líneas, de las que se ignoran las que comienzan con una almohadilla. Las líneas son de la forma:

keyword=value ó
keyword+value

pudiéndose añadir espacios en blanco entre keyword y el *igual* o el *más.* Todo lo que siga inmediatamente después del *igual* o el *más* es el valor o *value*, incluidos los posibles espacios en blanco que se incluyan. La versión con el igual configura la palabra clave con el valor indicado por *value*, mientras que la versión del signo suma concatena los valores indicados por *value* y los que haya predefinidos.

Cada grupo de palabras clave (keywords) define el comportamiento de **mirror**. El mínimo necesario para cada *package* incluye las siguientes:
"package", "site", "remote_dir" y "local_dir".

Aunque hay muchas palabras clave que pueden utilizarse, en la mayoría de los casos pueden obviarse y dejar que se tome el valor por defecto que se incluye desde **mirror**, aunque normalmente es casi seguro que habrá que poner los nuevos valores a los campos anteriormente citados.

*Configuración por defecto

Normalmente todas las configuraciones por defecto están en el fichero mirror.defaults que es cargado automáticamente antes que cualquier fichero de *packages*(ver Descripción), del cual se presenta a continuación el formato:

```
# Ejemplo mirror.defaults
package=defaults
    # Máquina LOCAL
    hostname=buitre.dit.upm.es
    # Directorio raíz a partir del cual colgarán todos los directorios
    local_dir=/public/
    remote_password=cualquiera@dit.upm.es
```

*Keywords

Lo que viene a continuación es una lista de las palabras clave disponibles y los valores que presentan por defecto al cargarse mirror. Para cambiar estos valores lo mejor es cambiar el fichero *mirror.defaults*

Las palabras clave están agrupadas en las siguientes secciones:

Palabras Clave requeridas
Ftp
Copia de ficheros
Atributos de ficheros locales
Borrado de ficheros
Compresión de ficheros
Partición de ficheros
Listados de directorios
Registros
Especial

Sólo se van a presentar las correspondientes a las tres primeras secciones por ser las de uso mas corriente, dejando para la documentación oficial al resto.

Palabras Clave requeridas:

Keyword	Por_defecto	Descripción
package	none	El nombre del *package* que va a ser copiado. Debería ser diferente del resto de los paquetes.
site	none	La dirección IP o el nombre del servidor remoto que vamos a copiar.
remote_dir	none	Directorio remoto que contiene lo que nos interesa. Ver también recurse_hard.
local_dir	none	Directorio local que albergará las copias.

FTP:

Keyword	Por_defecto	Descripción
remote_user	anonymous	Nombre de usuario en el servidor remoto.
remote_password	localuser@localhostname	Password a usar en el sitio remoto.
remote_account	none	Nombre de cuenta/password para usar desde el sitio remoto, después de entrar en él como anónimo.
remote_group	none	Si existe selecciona el remoto 'site group'.
remote_gpass	none	Si existe selecciona el remoto 'site gpass'.
timeout	40	Tiempo de timeout para peticiones FTP.
failed_gets_excl	none	Fichero de registro de mensajes de error, cuando falla FTP.
ftp_port	21	Número de puerto del demonio FTP remoto.
proxy	false	Ponerlo a *true* para habilitar el servicio FTP mediante proxy.
proxy_ftp_port	4514	Número de puerto del servicio de proxy del demonio FTP remoto.
proxy_gateway	internet-gateway	Nombre del servicio de proxy , posiblemente dado por la variable de entorno INTERNET_HOST
using_socks	false	Poner a *true* si se está utilizando una versión de Perl con SOCKS.
passive_ftp	false	Poner a *true* si se quiere utilizar la extensión PASV del protocolo FTP. Especialmente útil con firewalls, otros servidores proxy FTP, y la variable *using_socks.*
retry_call	true	Si la conexión inicial falla, lo reintenta UNA VEZ pasado UN minuto.
disconnect	false	Desconecta del sitio remoto al final del *package.* Normalmente sólo se desconecta si el siguiente *package* especifica un sitio distinto.

Copia de ficheros:

Keyword	Por_defecto	Descripción
get_patt		Registro de *pathnames* requeridos.
exclude_patt	none	Registro de los *pathnames* a ignorar.
local_ignore	none	Registro de los *pathnames* locales a ignorar. Útil para mantener sin modificar directorios locales restringidos.

get_newer	true	Copia el fichero remoto si es más reciente que el fichero local.
get_size_change	true	Copia el fichero si el tamaño es diferente del fichero local.
make_bad _symlinks	false	Si es *true* los enlaces simbólicos de ficheros no válidos(inexistentes) se copian también.
follow_local_symlinks	none	Registro de *pathnames* de enlaces simbólicos locales. En vez de tratarlos como enlaces se usan como si fueran los ficheros o directorios que referencian. Esto hace a los enlaces invisibles ante mirror.
get_missing	true	Realmente se transfieren ficheros. Si está puesto como *false*, solo se efectúan acciones de borrado y enlace.
get_file	true	Copiar ficheros. Si fuera *false* mirror trataría de enviar ficheros al servidor remoto.
text_mode	false	Si es *true*, todos los ficheros son transferidos en modo texto.

5.- PRUEBAS

Es difícil describir unas pruebas que no han podido ser realizadas por la indisponibilidad del servicio FTP en el laboratorio de programación, pero no es muy difícil imaginarse cómo debían de ser para comprobar que efectivamente el servicio **mirror** se ha configurado correctamente. Los pasos básicos podrían ser los siguientes:

- Instalarlo y comprobar que no hay errores de CRC.
- Seguir las instrucciones de este manual o de la documentación.
- Seleccionar un directorio en el que almacenar los contenidos que se desea reflejar del servidor original.
- Configurar **mirror** de la manera deseada.
- Ejecutarlo con la opción –n para ver simplemente qué pasaría en caso de que se ejecutase realmente.
- Tras comprobar que el programa no da ningún tipo de error a priori, ejecutarlo ya directamente (sin la opción –n).
- Esperar pacientemente a que se realice toda la transferencia
- Observación de posibles mensajes de error.
- Comprobación por comparación visual de que los ficheros almacenados en el directorio elegido para ello coinciden exactamente con lo que pretendíamos copiar.
- Y finalmente, felicitarnos por el éxito, si ha sido a la primera enhorabuena por la suerte que se ha tenido (¡qué difícil es hoy día que funcione algo a la primera!) y si ha sido tras varios intentos felicitarnos por nuestra tenacidad y confianza en lo que se estaba haciendo.

También convendría hacer alguna prueba para ver si las sucesivas actualizaciones se realizan correctamente, pero es obvia la comprobación: Observación de algún fichero que haya de renovarse y posterior cercioramiento de que la actualización ha tenido éxito.

6.- GESTIÓN DIARIA

La gestión diaria de **mirror** se facilita mucho con la correcta configuración de los *package files.* La automatización es realmente sencilla puesto que **mirror** se encarga completamente de buscar los ficheros que se han de actualizar, mediante la citada comparación de fechas y ficheros tanto del servidor remoto como del local.

El hecho de crear un **mirror** supone implícitamente una dedicación y un seguimiento del servidor del que se va a hacer copia. Por ello es necesaria una actualización periódica del contenido del servidor, lo que en sí no es casi nada costoso porque la transferencia real no es grande, en comparación con la inicial (sobre todo si es de un gran almacén de ficheros –repositorios- como sunsite, tucows, o muchos otros), si lo que realmente se pretende ser un espejo fiel del original. Realmente es de poca utilidad un **mirror** que no se actualiza porque si el usuario ve que el contenido no está actualizado con el original tenderá a dejar de utilizarlo para buscarse otro, o el original, con lo que el ofrecer este servicio perdería su sentido, al dejar de ser usado por no ofrecer utilidad al usuario. Y no es nada raro que los contenidos cambien con lo volátiles que son las versiones del software que se ofrece actualmente por Internet (y no digamos si lo que se pretende es hacer un **mirror** de servidores de antivirus, por ejemplo. ¿Qué utilidad podría tener un **mirror** en este caso si no se actualiza, por lo menos, una vez al mes?).

La tarea de configurar una frecuencia de actualización sería muy sencilla mediante la programación de acciones con *cron*, determinando la periodicidad deseada de copia.

A continuación presentamos una serie de consejos para un mejor aprovechamiento de **mirror:**

- Al añadir un nuevo *package*, probar si funciona previamente con la opción –n
- Si se van a añadir al directorio ficheros que no han sido transferidos por mirror (por ejemplo, copiados desde un CDROM), forzar a que la fecha de esos ficheros sea la misma que la que tienen en el CD para que funcione correctamente la comparación de fechas con los ficheros remotos (ver opción –T).
- Si se va a hacer un **mirror** de un sitio muy grande que cambia muy infrecuentemente, añadir "max_days=7" a la configuración después de la copia inicial. Así **mirror** sólo tiene que considerar los ficheros más recientes cuando haga la actualización. Entonces, una vez a la semana, o cuando sea necesario, ejecutar **mirror** con "–k max_days=0" para forzar una actualización completa.

También conviene seguir unas pequeñas reglas de *buen comportamiento* o cortesía, que pueden resumirse en los siguientes puntos:

- Observar cualquier restricción que hayan puesto los administradores del servidor del cual vamos a hacer el **mirror**. Normalmente estas restricciones se pueden ver nada más conectarse mediante FTP, quizá en un fichero de texto que se presente al visitante.
- Quizá haya que pedir permiso al administrador antes de hacer un **mirror** de su servidor. Algunos sitios piden ficheros *log* detallados, dado que **mirrors** no autorizados podrían generar tráfico del que no se tuviera conocimiento y por lo tanto falsearían sus estadísticas. Aparte puede haber serias razones legales por las que los **mirrors** sean indeseados.
- Sólo se debe realizar la copia fuera de las horas de trabajo tanto del sitio local como del remoto.
- Es probablemente inadecuado tratar de copiar un servidor más de una vez al día.
- Si se tiene hecho un **mirror** de algún sitio, avisar a los usuarios del sistema de ello para evitar que malgasten el ancho de banda tanto del sistema local como del remoto.
- Revisar de vez en cuando los posibles cambios en las restricciones del servidor remoto.