# DNS & IP's DINÁMICAS HOWTO

Jaime Pérez Crespo ( japecre at pantuflo.escet.urjc.es)

Última actualización: 11 de Julio de 2003

## Motivación

Hoy en día el tener en casa una conexión de alta velocidad, con una dirección IP (Internet Protocol) fija asociada a dicha conexión, se ha convertido en algo habitual. Y por tanto, más habitual aún es comprobar como usuarios de este tipo de conexiones aprovechan el tener una IP fija para ofrecer ellos mismos ciertos servicios al resto de la comunidad internauta, como bien pueden ser servidores FTP (File Transfer Protocol), de correo, WEB, e incluso otros.

Sin embargo la masiva utilización de este tipo de líneas, está provocando un uso desmesurado de direcciones IP fijas, lo cual hace que el de por si limitado mapa de direcciones IP que disfrutamos aún hoy en IPv4 se vea muy saturado. Es por ello que muchos ISP (Internet Service Provider) han optado para su modelo de negocio por un sistema de asignación dinámica de direcciones IP mediante DHCP (Dinamic Host Configuration Protocol) u otros protocolos; y muchos otros ya están pensando en pasarse a este modelo para ahorrar costes y direcciones IP en definitiva. Esto dificulta la tarea de aquellos que quieran montar un servicio público en su máquina, al no tener una IP fija sobre la que dar el servicio, si bien, gracias al sistema de DNS (Domain Name System) que hoy por hoy es el pilar de la red, pueden asociar un nombre a su servicio de forma permanente, gratuita, y sencilla, sin importar el tipo de conexión que se utilice.

El objetivo de este mini HOWTO es por tanto facilitar la tarea a aquellos que, bajo GNU/Linux, quieren montar un servicio disponible en Internet, bajo una conexión que les proporciona una dirección IP dinámica o semidinámica. La idea general es asociar un nombre DNS, mediante uno de los múltiples servicios gratuitos que existen en la red, a nuestra máquina, por medio de un DNS updater. A continuación detallamos el proceso.

## Primeras consideraciones

Obviamente, el servicio que prestemos es lo de menos importancia. Durante este HOWTO, los ejemplos se pondrán bajo una plataforma i686 trabajando con Red Hat GNU/Linux 7.3, si bien serán extrapolables a otras plataformas GNU/Linux, en mayor o menor medida.

Lo primero que necesitamos es un servicio que permita asociar un registro DNS con una dirección IP, para lo cual, no nos vale cualquiera. Hay que tener en cuenta que lo que nos interesa es disponer de un nombre fijo, asociado en todo momento y de forma actualizada, a la dirección IP que tiene la máquina en la que ofrecemos el servicio. Por tanto, deberemos elegir un servicio de DNS que nos proporcione un pequeño actualizador, un sencillo programa que se conecte con el servicio y actualice automáticamente la IP asociada al registro DNS cada vez que detecte un cambio. En este caso, vamos a utilizar los servicios de http://www.no-ip.com/, ampliamente usados y reconocidos por la comunidad internauta, y con todas las características que nos permitirán explicar este HOWTO con todo lujo de detalles.

Nos registramos en el servicio, y elegimos el nombre que queremos asociado con nuestra máquina. Podemos utilizar un nombre que sea descriptivo, de entre los que nos proporcionan, del servicio que queremos dar en nuestro PC. Por ejemplo, si queremos montar un servidor FTP, podemos elegir un dominio para nuestro nombre como serveftp.com y darle un nombre que nos identifique, como puede ser el nombre de nuestra máquina o host. Un ejemplo: mordor.serveftp.com

Una vez hayamos completado el proceso, tendremos que descargar el cliente que se nos proporciona para actualizar automáticamente el registro DNS. Ya descargado a nuestra máquina, lo descomprimimos como siempre:

```
[icarus@mordor]$ tar zxvf noip_updater_v1.6.tar.gz
noip_updater_v1.6/
noip_updater_v1.6/Makefile
noip_updater_v1.6/COPYING
noip_updater_v1.6/README.FIRST
noip_updater_v1.6/no-ip.sh
noip_updater_v1.6/no-ip.conf.sample
noip_updater_v1.6/noip.c
```

```
noip_updater_v1.6/i386binaries/
noip_updater_v1.6/i386binaries/noip-for-libc6
```

Esto nos descomprime todos los archivos del cliente en un directorio llamado noip_updater_v1.6 bajo el directorio en el que nos encontremos a la hora de descomprimir. Como siempre, es conveniente leerse dos veces los archivos de texto que acompañan a cualquier programa, como el README.FIRST en este caso.

Procedemos ahora a instalar correctamente el cliente, utilizando para ello make, make conf y make install (é ste último estando identificado como root):

```
[icarus@mordor]$ cd noip_updater_v1.6
[icarus@mordor]$ make
gcc -O2 -g -Dlinux noip.c -o noip
[icarus@mordor]$ make conf
./no-ip.sh
Greetings!

The no-ip dynamic DNS system requires a configuration file.
The answers to the next five questions will build one.

What is your email address for no-ip.com? (name@email.addr)
> nuestro_correo@dominio.com
What is the password for the login account? (password)
> nuestro_password
Are you using a groupname instead of a host.domain.name? (Y/N)
Answer 'N' if you do not understand the question.
> N
What is the host.domain.name you are using? (host.no-ip.com)
> mordor.serveftp.com
Do you have a router which uses NAT/masquerading? (Y/N)
> N
Which device is your net connection? (one of: lo eth0 eth1 )
> eth0
Configuration file [./no-ip.conf] created.
```

El comando make compila el cliente de no-ip a partir del Makefile que hemos descomprimido previamente. Una vez tenemos el binario, pasamos a configurar el cliente con make conf. Indicamos por este orden, según se nos pregunta, la dirección de correo con la que nos registramos en el servicio de no-ip; la contraseña que tenemos en dicho servicio; N si hemos registrado el nombre de nuestra má quina; el nombre completo que hemos asociado, en nuestro caso, el mostrado; N de nuevo si no tenemos un router dando servicio a varias máquinas (en multipuesto) utilizando NAT (Network Address Translation); y finalmente el nombre del adaptador de red que da conexión a la má quina, en la mayoría de los casos, eth0.

Procedemos a terminar de instalar el cliente en nuestro sistema. Este paso requiere que nos identifiquemos como superusuario (root) con lo cual, no podremos llevarlo a cabo en aquellos sistemas en los que tan solo tengamos una cuenta de usuario sin privilegio alguno. En cualquier caso, no es imprescindible llevar a cabo este paso para poder utilizar el cliente, ya que en este punto, ya se puede usar simplemente situá ndonos en el directorio donde descomprimimos el cliente, y tecleando ./noip -c no-ip.conf

```
[icarus@mordor]$ su
Password:
[icarus@mordor]$ make install
cp noip /usr/local/bin/noip
cp no-ip.conf /usr/local/lib/no-ip.conf
chmod 600 /usr/local/lib/no-ip.conf
```

Como podemos ver, simplemente se ha copiado el binario a un directorio que forma parte del path del sistema, y el archivo de configuración al directorio donde nuestro cliente busca por defecto dicho archivo. Por último se han quitado permisos al archivo de configuración para evitar una manipulación malintencionada por parte de algún usuario de la má quina. Hecho esto, tenemos nuestro cliente listo para ejecutar, con tan solo teclear en una shell el comando noip.

## Cuestiones avanzadas

Si bien ya tenemos todo lo estrictamente necesario para poder empezar a ofrecer nuestro servicio, sería deseable que nuestro registro DNS estuviese siempre actualizado, sin necesidad alguna de que nosotros tengamos que intervenir arrancando el cliente. Lo que vamos a hacer ahora pues, es crear un nuevo servicio del sistema para correr nuestro cliente como un demonio, que se podrá arrancar con cada inicio y así mantendrá la DNS actualizada siempre.

Comenzamos por crear el servicio. Nos identificamos como root, y nos dirigimos a /etc/init.d. Una vez allí, creamos un nuevo archivo, que será un script de shell asociado a nuestro nuevo servicio, y lo editamos.

```
[icarus@mordor]$ su
Password:
[root@mordor]# cd /etc/init.d
[root@mordor]$ vi noip
```

En este caso usamos vi por ser un editor ampliamente extendido y presente en casi todos los sistemas. No nos extenderemos aquí en explicar el uso de este editor. Sirva decir que creamos el archivo con vi noip tras lo cual en el propio editor, pegaremos el siguiente contenido:

```
#!/bin/sh
case "$1" in
        start)
                noip
                echo "Cliente no-ip arrancado."
        ;;
        stop)
                killall -TERM /usr/local/bin/noip > /dev/null
                echo "Cliente no-ip parado."
        ;;
        status)
                lines="`ps aux | grep -c "noip"`"
                if [ "$lines" -gt 1 ]
                then
                        echo "Cliente no-ip arrancado."
                else
                        echo "Cliente no-ip parado."
                fi
        ;;
        *)
                echo "Uso: $0 {start|stop|status}"
                exit 1
esac
exit 0
```

Una vez hemos pegado el contenido, y guardado el archivo, procedemos a dar permisos adecuados al nuevo script de shell para poder utilizarlo:

```
[root@mordor]# chmod 755 noip
```

Hecho esto, ya podemos utilizar nuestro nuevo servicio de sistema. En nuestro caso hemos dado soporte a tres opciones distintas para nuestro servicio, start, stop y status, que nos permitirán arrancar, detener, y consultar el estado del demonio respectivamente. A continuación se muestra un ejemplo de cómo accederemos al servicio:

```
[root@mordor]# service noip
Uso: /etc/init.d/noip {start|stop|status}
[root@mordor]# service noip status
Cliente no-ip parado.
[root@mordor]# service noip start
Cliente no-ip arrancado.
[root@mordor]# service noip status
Cliente no-ip arrancado.
[root@mordor]# service noip stop
Cliente no-ip parado.
[root@mordor]# service noip status
```

```
Cliente no-ip parado.
```

Sólo resta configurar el nivel de ejecución que deseemos para que nuestro nuevo demonio se arranque cada vez que iniciemos ese nivel (o niveles). Por ejemplo, para una máquina que actúa como servidor las 24 horas y que no utiliza entorno gráfico, lo usual es que arranque en nivel de ejecución 3. Si solemos arrancar la máquina con el entorno de ventanas que prefiramos, usaremos nivel de ejecución 5. Para comprobar en qué nivel de ejecución estamos en cualquier momento, podemos utilizar el siguiente comando:

```
[icarus@mordor]$ grep initdefault /etc/inittab | awk -F: '{print $2}'

5
```

Una vez conocido el nivel de ejecución, podemos usar alguna herramienta que nos proporcione nuestra distribución de GNU/Linux para editar el inicio. En nuestro caso, utilizamos ksysv, que mediante una sencilla GUI, permite arrastrar servicios de la lista de servicios disponibles a los niveles de ejecución que deseemos, precisando si en ese nivel de ejecución queremos parar o arrancar el servicio. No nos extenderemos más en este punto, ya que la variedad de herramientas disponibles para este propósito hacen imposible explicar todas las formas de hacer esto.

Y esto es todo. Llegados aquí nuestro sistema debería tener un nombre de escala mundial, asociado permanentemente, sea cual sea nuestra conexión, y actualizado puntualmente con cada cambio. Espero que este tutorial haya servido de ayuda a aquellos que no saben bien como ofrecer un servicio tal como un FTP al resto de usuarios de la red.

## Enlaces Interesantes

No-ip: www.no-ip.com
Virtual Services HOWTO: www.ibiblio.org/mdw/HOWTO/Virtual-Services-HOWTO.html

## Licencia

Este documento público puede ser distribuido y/o traducido libremente, siempre y cuando en el documento resultante se incluyan los datos del autor indicados en la cabecera, esta sección de licencia, y la url del documento original:
http://pantuflo.escet.urjc.es/~japecre/trabajos/independientes/docs/dnshowto.php?idioma=es