Analizamos la vanguardia de los navegadores alternativos: Neoplanet y Opera

internet ONLINE

www.prensatecnica.com

LA REVISTA MÁS PRÁCTICA PARA PROFESIONALES DEL MUNDO DE INTERNET

AÑO 2 • NÚMERO 20 • PC • MAC
PORTUGAL 990 ESC (CONT)
5,98 €

995 Ptas

INCLUYE CD ROM PC-MAC

CHAT
Todo lo que hay que saber sobre el IRC

SOFTWARE
Subir ficheros a la Red y automatización de tareas

MAC
Aplicaciones de entretenimiento

FIRMA
La salud del comercio electrónico en España

DISEÑO WEB
Aprenda HTML con Homesite

ÉTICA EN LA RED
La moral en las páginas de Internet

CD ROM

- Últimas versiones de navegadores
- Demo de Fireworks 2
- Homesite 4.0
- Demo de LapLink Pro
- Shareware
- MIRC 5.41 en Español
- Especial Download

Y muchas cosas más...

Microsoft Internet Explorer 5

Netscape Communica

Comparativa

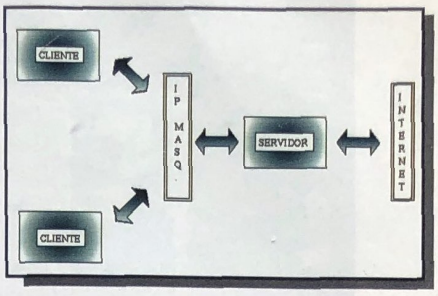
Netscape 4.51 Vs Explorer 5.0

Prensa Técnica
8 413042 087830 00020

Santiago Romero Iglesias
sromero@arrakis.es

IP-MASQUERADE (II)

Conexión y configuración de Internet desde Linux

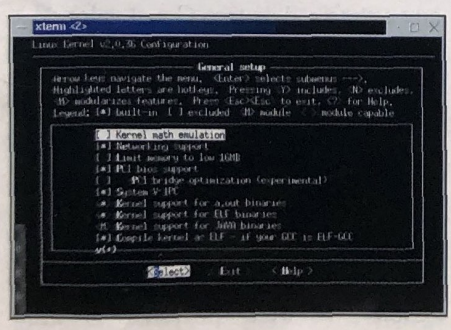


Mediante la técnica de IP-Masquerade en Linux es posible tener acceso a Internet en ordenadores conectados a una red local donde al menos uno de ellos sí que posea conexión al exterior (el ordenador pasarela). En esta entrega veremos cómo configurar un servidor de Internet y sus clientes con IP-Masquerade.

En estos tiempos donde se jubilan PC's con procesadores 386 o 486 es muy habitual tener en casa pequeñas redes de dos ó mas ordenadores (generalmente 2 ordenadores), teniendo el más potente de ellos su modem propio y acceso a Internet. Sería muy interesante «compartir» el modem y poder tener un acceso simultáneo a Internet por ambos PC's, compartiendo el ancho de banda del modem. Esto es posible en Linux gracias a la técnica de *IP MASQUERADE*, y en esta entrega veremos cómo se instala y configura este servicio, tanto para el servidor, un Linux, como para el cliente, que podrá ser un Linux, Windows, o cualquier otro S.O, tal y como se ilustra en la figura 1.

Mediante IP Masquerade es posible tener acceso a Internet en una red vía LAN

IP Masquerade es una técnica que todavía está en desarrollo en algunos kernels, pero que funciona perfectamente y está incluido en el kernel de Linux desde la versión 1.3.0. Mediante IP-Masquerade, cualquier ordenador (con cualquier S.O.) conectado a un servidor (en nuestro caso uno de los ordenadores de la red con un simple modem) podrá hacer HTTP, FTP, TELNET, IRC o incluso jugar a Quake como si estuviese conectado a Internet, gracias al servidor Linux. Para explicar la técnica de IP Masquerade usaremos un ejemplo real: una red de 2 ordenadores con sendas tarjetas de red compatibles NE2000, uno de ellos con un modem de 33k6, y Linux, y el otro un cliente Windows 95 con Netscape instalado.



Make menuconfig.

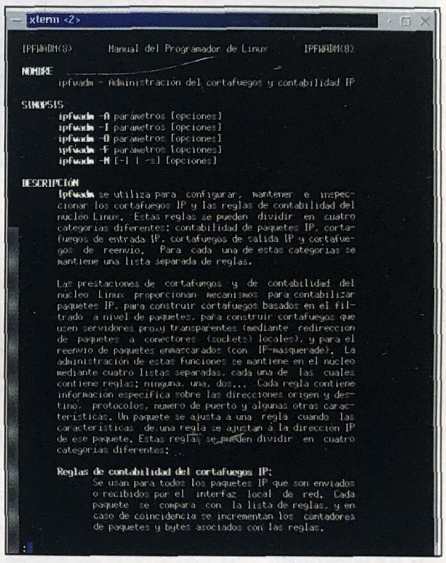
Configuración en la Red

Lo primero de todo es la correcta configuración de las tarjetas de red. Existen diferentes páginas y HowTo's que cubren esta materia, pero sin duda alguna los mejores tutoriales los podemos encontrar en LuCAS (http://*lucas.ctv.es*), aunque nosotros usaremos la configuración de red utilizada en nuestro último número.

Como se vio en la anterior entrega, para la configuración de las tarjetas de red es necesario recompilar el kernel con soporte para la tarjeta de la que se disponga, así como habilitar las opciones necesarias para IP - Masquerade. Debido a los diferentes emails recibidos preguntando sobre la compilación del kernel se ha optado por realizar una completa guía de compilación en el último apartado del artículo (le recomendamos ir primero a dicha sección si es la primera vez que va a realizar este proceso).

Antes de pasar a configurar IP Masquerade es imprescindible configurar la red

En el kernel debemos habilitar el soporte para redes, soporte TCP/IP, etc., así como soporte para la tarjeta de red que se vaya a utilizar. Si ésta es una ne2000 PCI al rearrancar el nuevo kernel será arrancada y configurada automáticamente, mientras que si es ISA, tal vez debamos indicar los parámetros para la tarjeta (dirección I/O y similares) en el fichero /etc/conf.modules, o utilizando *modconf* en Debian, tal y como se explica en los diferentes HowTo's sobre el tema o en la página de Monkiki. Aquí vamos a usar un ejemplo de red de 2 ordenadores: el servidor Linux (con IP=192.168.5.1) y el cliente Windows (con IP=192.5.168.2). En nuestro ejemplo ambas tarjetas son PCI (no necesitan configuración adicional, para tarjetas de otro tipo consultar los diferentes HowTo's, pasarse por es.comp.os.linux, o por nuestra última entrega).



Página man de ipfwadm.

Una vez tengamos la red correctamente configurada y testeada con ping, telnet y ftp funcionando entre las diversas máquinas, puede procederse a la configuración de IP Masquerade.

Activar IP Masquerade

Para activar IP Masquerade se debe recompilar el kernel con las siguientes opciones activadas (pueden cambiar según la versión del kernel, pero el nombre será similar):

Allow experimental code (CONFIG_EXPERIMENTAL) -> permite el uso de código experimental (en algunos kernels puede ser necesario).

Enable loadable module support (ALLOW_MODULES) -> permite carga de módulos

Networking support

Si no termina de entender el significado del cero en la dirección de red, piense en él como en un comodín, de tal forma que le evita la configuración de IP Masquerade para todas las diferentes máquinas de la red (también es posible hacerlo de forma individual):

ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.5.2/24 -D 0.0.0.0/0
ipfwadm -F -a m -S 192.168.5.3/24 -D 0.0.0.0/0
(etc..)

En los nuevos kernels 2.2.x el paquete *ipfwadm* ha quedado obsoleto (desde que empezó la gran evolución de los kernels de desarrollo lo es) y su sustituto es IPCHAINS, que podemos encontrar en las siguientes direcciones:

http://rpmfind.net/linux/RPM/suse/6.0/n1/ipchains-1.3.8-6.i386.html
http://www.adelaide.net.au/~rustcorp/linux/ipchains

Los clientes Win9x se configuran desde el Asistente de Conexiones y el Panel de Control

Ipchains ha sustituido a ipfwadm aunque mantiene una sintaxis similar (consultar el IPChains Howto para ver las posibles opciones del mismo), aunque en el mismo paquete de ipchains viene un script que simula los diferentes parámetros de ipfwadm para que no tengamos que aprender el nuevo formato (simplemente se llama a ese script con los parámetros que usabamos para ipfwadm y él se encarga de llamar a ipchains transformando los parámetros al formato entendible por este último).

Nuestras anteriores reglas para IP masquerade quedaría con ipchains de la siguiente manera:

ipchains -P forward DENY
ipchains -A forward -j MASQ -s 192.168.5.0/24 -d 0.0.0.0/0

o también:

ipchains -P forward DENY
ipchains -A forward -j MASQ -s 192.168.5.0/24 -d any/0

Tras esto sólo queda la configuración de los clientes Windows o Linux para disfrutar de conexión a Internet en el resto de máquinas de la red. Todas las máquinas de la red tendrán el mismo IP para el exterior, pero si tenemos activado el *identd* de Linux se podrá identificar al usuario de cada máquina por su login.

Configuración del cliente

Si el cliente es un Windows 9x con la tarjeta de red correctamente instalada y configurada, sólo tendrá que ejecutar el asistente de conexiones (Menú Inicio, Aplicaciones, Sistema, Internet, Conectarse a Internet), y seleccionar *configuración manual, conectar usando una red LAN*, y *No usar proxy* (ver figura 3).

También será necesario ir a Panel de control, Red (propiedades), Propiedades de TCP/IP de la tarjeta de red en cuestión, y allí seleccionar Puerta de Enlace y añadir allí la dirección IP del servidor Linux. En la pestaña de DNS se deberá *Activar DNS* e introducir los mismos DNSs que se utilicen para resolver nombres en la máquina Linux (en el fichero /etc/resolv.conf, aunque también podría utilizar cualquier otro DNS), así como dotarlo del hostname de la máquina Windows en la casilla *Nombre de host* (figura 4). Tras esto deberá reiniciar el cliente Windows (no así el Linux, aunque deberá insertar los módulos o tal vez desee rearrancar para ver si incluyó correctamente las entradas en el fichero /etc/rc.d/rc.local).

La mejor manera de probar IP Masquerade es conectar mediante el servidor Linux y abrir Netscape en el cliente Windows 9x, e introducir una dirección para tratar de navegar con normalidad.

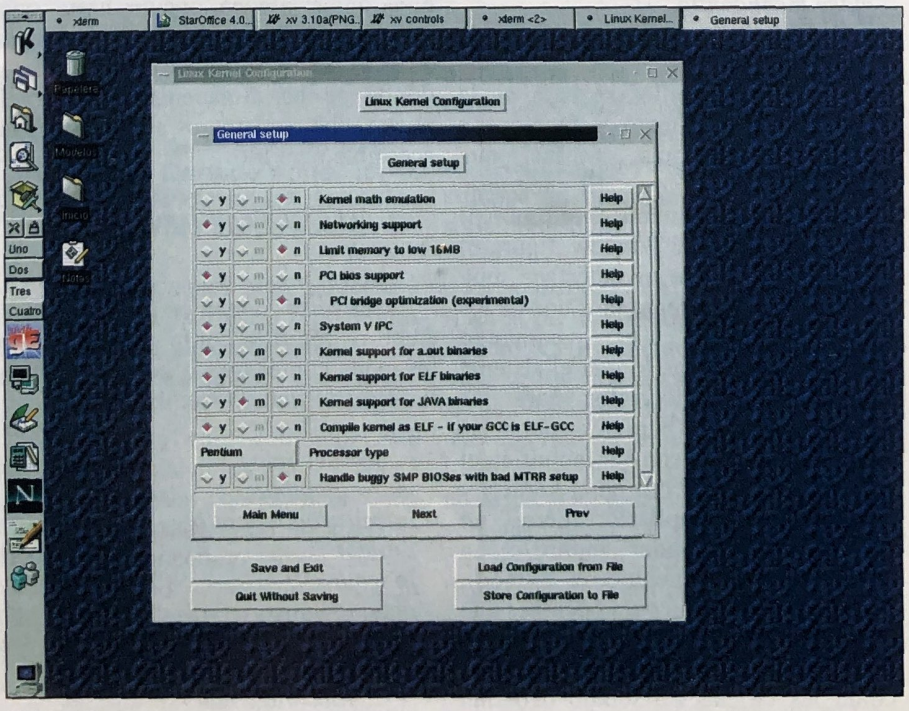
Para configurar clientes Linux, tan sólo deberá utilizar control-panel o editar /etc/sysconfig/network o /etc/sysconfig/network-scripts/ipcfg-eth0 para indicar el GATEWAY (puerta de enlace) correcto, especificando la dirección IP dentro de la red del Linux servidor. El DNS y domain del ISP (como se explicó en nuestra primera entrega) deben ir en el fichero /etc/resolv.conf, reiniciando el sistema (o los servicios implicados).

Compilación del Kernel

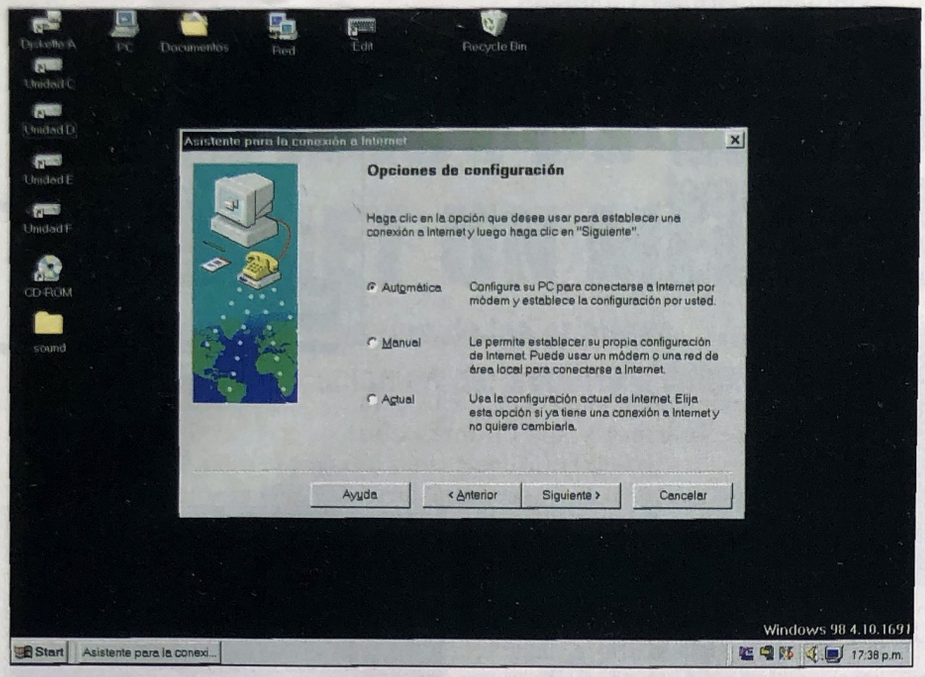
La compilación del kernel es un paso muy habitual y sencillo en la configuración de Linux, y aunque en principio pueda parecer un proceso complicado y que requiera conocimientos de programación, aunque nada más lejos de la realidad, ya que en Linux es un paso casi automatizado en su totalidad.

La compilación del kernel es un proceso muy habitual, sencillo y automatizado

Compilar el kernel no tiene ningun peligro si se realiza una copia de seguridad del kernel antiguo en una unidad de tipo msdos o en un diskette cualquiera y además en la misma unidad/disqete se copia el programa de MSDOS *loadlin.exe*, incluido en la mayoría de Cds de Linux, y que permite arrancar Linux desde MSDOS a partir de un kernel copia de seguridad. Si se estropeara algo en el kernel, puede arrancarse con loadlin usando el viejo, y recompilar, deshacer cambios, copiar el kernel viejo sobre el



Make xconfig.



Asistente de conexiones de Win9x.

(CONFIG_NET)-> permite soporte de red.

Network firewalls (CONFIG_FIREWALL)

IP: firewalling (CONFIG_IP_FIREWALL) -> permitir uso de firewalls o cortafuegos.

TCP/IP networking (CONFIG_INET) -> Activar el uso de TCP/IP

IP: forwarding/gatewaying (CONFIG_IP_FORWARD) -> Permitir IP forwarding.

IP: masquerading (CONFIG_IP_MASQUERADE) -> Permitir IP-Masquerade.

IP: ipautofw masquerade support (CONFIG_IP_MASQUERADE_IPAUTOFW)

IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) -> Estas 2 últimas opciones estarán disponibles en los kernels a partir de la 2.0.30, y son de activación recomendada.

IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG) -> Recomendado.

Dummy net driver support (CONFIG_DUMMY) -> Permitir soporte de red dummy.

Aparte de estas opciones deberá incluir las de su tarjeta de red, los periféricos conectados a su PC, etc... como en cualquier compilación del kernel, y realizar el *make zImage*. También deberá crear los módulos mediante *make modules* y *make modules_install*. Con esto, aparte de los módulos seleccionados por el lector en las opciones del kernel, se crearán los módulos necesarios para IP-Masquerade, que permitirán FTP, IRC, Real Audio, etc. Para que estos se carguen en memoria en cada arranque se pueden incluir en cualquier fichero de arranque, como /etc/rc.d/rc.local (o similares, según distribuciones), mediante unas líneas como las siguientes (y dependiendo de los módulos o servicios que se quieran activar):

/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
/sbin/modprobe ip_masq_quake

El total de módulos disponibles (version del kernel 2.2.1) son:

/lib/modules/2.2.1/ipv4/ip_masq_cuseeme.o
/lib/modules/2.2.1/ipv4/ip_masq_ftp.o
/lib/modules/2.2.1/ipv4/ip_masq_irc.o
/lib/modules/2.2.1/ipv4/ip_masq_quake.o
/lib/modules/2.2.1/ipv4/ip_masq_raudio.o
/lib/modules/2.2.1/ipv4/ip_masq_user.o
/lib/modules/2.2.1/ipv4/ip_masq_vdolive.o

Tras copiar el nuevo kernel en /boot y preparar el sistema para que arranque con él, deberemos ejecutar lilo -v para actualizar los cambios, y ejecutar la siguiente línea para activar IP Masquerade:

echo "1" > /proc/sys/net/ipv4/ip_forward

Para usuarios de Redhat es también necesario editar el fichero /etc/sysconfig/network y activar la opción de IP_FORWARDING mediante *FORWARD_IPV4=true*.

Una vez activado el soporte IP Forwarding se debe configurar dicho soporte mediante el paquete *ipfwadm* (ver figura 2), que deberemos tener instalado (viene de serie en prácticamente todas las distribuciones). Este programa es el encargado de reenviar los paquetes a la máquina apropiada dentro de la red (figura 1), siendo pues el mediador entre las máquinas clientes y la servidora en cuanto al reenvío de paquetes IP.

Mediante los diferentes módulos de IP Masquerade obtenemos acceso a HTTP, FTP, telnet e IRC

Para la configuración de las reglas de ipfwadm tan sólo hay que ejecutar este programa con los parámetros apropiados en la máquina servidora, normalmente una variante de las líneas siguientes:

ipfwadm -F -p deny
ipfwadm -F -a m -S <red>/<X> -D 0.0.0.0/0

En las 2 líneas anteriores se deben sustituir los parámetros entre <> por los valores correctos para su red. El primer parámetro se corresponde con la dirección IP de su red, con un cero en el lugar de la dirección IP que es variable. Es decir, si las máquinas de su red son 192.168.5.1, 192.168.5.2, .3, etc... Entonces su dirección de red es 192.168.5.0 (un cero en la parte de la dirección IP susceptible de cambiar). El valor del segundo parámetro (X) depende de la máscara de subred (del tipo de red), según la siguiente tabla:

Máscara de red	X	Tipo de red
255.0.0.0	8	Clase A
255.255.0.0	16	Clase B
255.255.255.0	24	Clase C
255.255.255.255	32	PPP

La máscara de subred de nuestra red 192.168.5.0 es 255.255.255.0, con lo que es una red de tipo C y el valor de X es, pues, de 24. Por lo tanto al final del fichero /etc/rc.d/rc.local (según distribuciones, o incluso se puede ejecutar en línea de comandos cuando sea necesario) habría que añadir las siguientes líneas:

ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.5.0/24 -D 0.0.0.0/0

nuevo, etc. El uso de loadlin es muy sencillo:

loadlin zimage root=/dev/hda5: rw

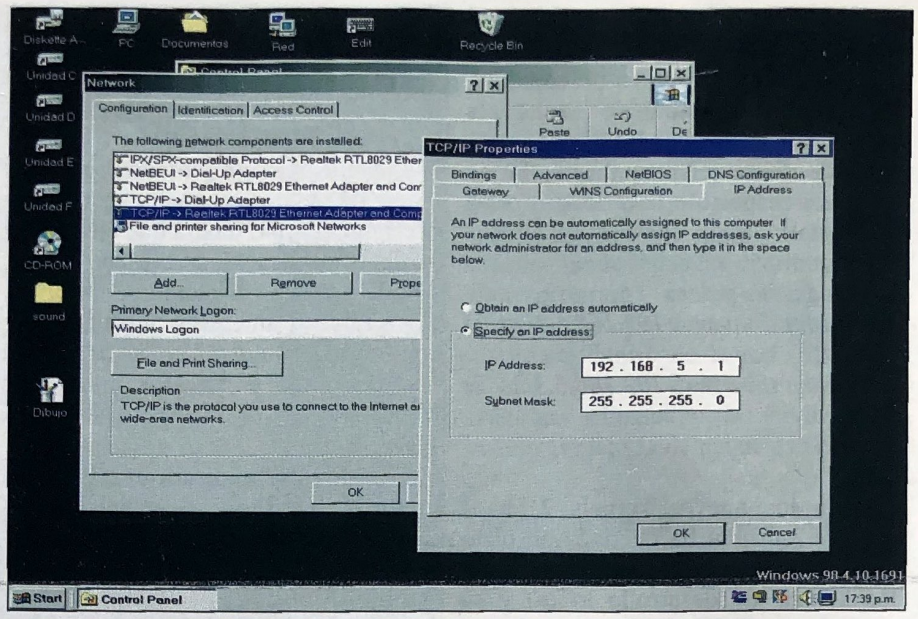
Estando en modo msdos se ejecuta el programa (loadlin) pasándole el nombre del kernel a arrancar (zimage) e indicándole donde está el sistema de ficheros que debe montar en el root (/) con el parámetro (root=/dev/hdxx, indicando xx la partición a arrancar). El rw le indica que monte dicho sistema de ficheros para lectura (letra r) y escritura (w).

Con loadlin podemos arrancar Linux desde MSDOS

Para compilar el kernel se deben de tener instalados los fuentes y ficheros de cabecera del kernel. En RedHat 5.2 (a título de ejemplo) estos paquetes se llaman *kernel-2.0.36-0.7.i386.rpm*, *kernel-headers-2.0.36-0.7.i386.rpm* y *kernel-source-2.0.36-0.7.i386.rpm* (variando la versión según la que tengamos instalada). Todo el proceso de compilación debe hacerse en el directorio */usr/src/linux*, y se basa en una serie de comandos y configuraciones que se realizan de una forma muy rápida y sencilla. Para modificar las opciones que debe llevar el nuevo kernel, y siempre dentro del directorio /usr/src/linux, podemos optar entre 3 formas de configuración, cada una de las cuales más visual que la anterior. Los 3 comandos son:

make config
make menuconfig
make xconfig

El primero de ellos se basa en la consola en modo texto y funciona a base de preguntas a las que contestaremos *y/n/m*. El segundo funciona a base de menús (ver figura 5) y el tercero (para X Window, figura 6) se realiza mediante un interface muy visual en Tcl/Tk. Elegiremos uno de los tres tecleando el comando correspondiente, tras lo que el sistema nos preguntará el valor para las diferentes opciones para el kernel, a las que contestaremos con SÍ (incluir en el kernel), NO (no incluir en el kernel) y MÓDULO (compilarlo pero no incluirlo, para poder incluirlo en cualquier otro momento con insmod). Los módulos se usan para opciones que no se utilicen habitualmente, como por ejemplo una unidad zip, pero cosas como las tarjetas de sonido deberían incluirse directamente en el kernel



Configuración de la tarjeta de red.

(ocupando memoria). Las opciones se agrupan en subsecciones generales, de red, de sonido, etc.

Tras elegir las opciones salimos grabando (Save and Exit) y tecleamos los comandos para comenzar la compilación:

make dep clean zImage

Al final del proceso (unos 5 minutos en un P200) saldrá algo similar a:

System is 500Kb size.
System copied in /usr/src/linux/arch/i386/boot/zImage

Si el kernel excede de 512Kb se nos notificará un error y deberemos realizar la compilación con *make bzImage* o bien quitar alguna opción hasta que entre dentro de este rango (máx. 512Kb). El último mensaje nos indica dónde se ha almacenado el kernel nuevo, que copiaremos a un lugar más accesible para lilo:

cp arch/i386/boot/zImage /boot/zImage

Tras esto debemos modificar el fichero /etc/lilo.conf para que se arranque con el nuevo kernel. Veamos un sencillo fichero /etc/lilo.conf:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux
image=/boot/zImage-2.2.1
        label=linux
        root=/dev/hda5
        initrd=/boot/initrd-2.2.1-0.7.img
        read-only
other=/dev/hda1
        label=win
        table=/dev/hda
```

Como puede verse, su estructura es muy sencilla. Las primeras líneas (boot, map, install) son indicativas para el lilo. Con *timeout = 50* hace que lilo espere en el arranque 5 segundos para que el usuario elija el S.O. con que arrancar. Pasados estos arrancará el que haya por defecto (opción default). Por último *image* indica con qué kernel arrancar, y esta es la línea a cambiar por /boot/zImage. Tras esto se ejecuta lilo -v en la línea de comandos para actualizar los cambios realizados en el fichero (obligatorio).

Por último, para compilar los módulos existentes se deben ejecutar los siguientes comandos:

make modules modules_install
depmod -a

Tras esto se puede rearrancar y comprobar el kernel que se está usando con un *uname -a*:

Linux compiler.linux.es 2.2.1 #4 mar ene 8 21:12:22 i586 unknown

En resumen

En esta entrega hemos aprendido a configurar IP Masquerade para así disfrutar de conexión a Internet en toda nuestra red de ordenadores (y todo por el mismo precio), tanto para clientes Windows como Linux, así como a compilar el kernel (ideal para aquellos que todavía se están iniciando en el mundo Linux) la sencilla guía de compilación del kernel incluida al final del artículo.