

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

Tutorial desarrollado por: [Alejandro Perez García 2003-2005](#), nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.

Si te gusta lo que ves, **puedes contratarme** para impartir **cursos presenciales** en tu empresa o para ayudarte en proyectos (Madrid).

Contacta: alejandropg@autentia.com.



Descargar este documento en formato PDF [debianQemu.pdf](#)

Software Libre y Linux

Servidores Linux, ERP, CRM, CMS
Servicios para empresas

CCS

Centro de Cálculo de Sabadell 1ª
empresa española de software

Módulos de automatización

Periferia y comunicación de PLCs
en el entorno Siemens.

Nexedi ERP5

Consulting and development
services Open Source ERP5, CRM,
CPS, Zope

Anuncios Goooooogle

Anunciarse en este sitio

Como instalar Windows XP sobre Debian GNU/Linux con QEMU

Creación: 06-10-2005

Última modificación: 28-12-2005

Índice de contenidos

- [1. Introducción](#)
- [2. Entorno](#)
- [3. Instalamos los paquetes necesarios](#)
 - [3.1. Compilación de qemu con aceleración](#)
- [4. Configuración de la aceleración de qemu con qemu](#)
- [5. Creación de una imagen de disco](#)
- [6. Instalación de Windows XP en qemu](#)
- [7. Comprobación de la aceleración con qemu](#)
- [8. Configuración de red](#)
 - [8.1. Configuración avanzada de red](#)
- [9. Conclusiones](#)
- [10. Sobre el autor](#)

1. Introducción

A menudo trabajando con Linux no nos queda más remedio que ejecutar aplicaciones que sólo están disponibles en Windows. Para esto tenemos varias soluciones:

- Reiniciar la máquina y arrancar con una partición de Windows.
- Usar wine, o algún producto similar. Este tipo de herramientas permite ejecutar aplicaciones de forma directa sobre Linux,

sustituyendo las llamadas al sistema. El problema de estas herramientas es que no son 100% compatibles y puede haber muchas aplicaciones Windows que no nos funcionen.

- Usar emuladores o máquinas virtuales. Este tipo de herramientas simulan un ordenador completo, pudiendo instalar cualquier sistema operativo en su interior. Además tienen la ventaja de que también nos permiten simular distintas plataformas.

En este tutorial vamos a ver una herramienta del tercer tipo. En concreto veremos QEMU (<http://fabrice.bellard.free.fr/qemu/>), una de las mejores y más rápidas herramientas de emulación que podemos encontrar en el mundo open source.

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Ahtex Signal X-9500M (Centrino 1.6 GHz, 1024 MB RAM, 60 GB HD).
- Sistema Operativo: GNU / Linux, Debian Sid (unstable), Kernel 2.6.12, KDE 3.4
- udev 0.070-2
- qemu 0.7.2

3. Instalamos los paquetes necesarios

Para instalar de forma rápida qemu podríamos hacer:

```
# apt-get -u install qemu
```

Pero en este caso no vamos a coger esta opción. El motivo es porque si instalamos qemu de los paquetes oficiales no vamos a tener disponible la aceleración de la emulación, y os puedo asegurar que es algo altamente recomendable. Por esto en esta ocasión vamos a optar por compilar nosotros mismos qemu (actualmente es la única forma de conseguir la aceleración).

Emulator	Typical application slow downratio compared to native
QEMU	between 5 and 10
QEMU + KQEMU accelerator	between 1 and 2

Aunque el hecho de tener que compilar nosotros qemu sea más trabajoso, por la tabla podéis ver que va a compensar, porque los tiempos de ejecución que se consiguen en la emulación mejoran considerablemente.

3.1. Compilación de qemu con aceleración

Atención!!! Sólo deberías hacer lo que se indica en esta sección si NO instalaste el paquete oficial Debian de qemu (apt-get -u install qemu).

En primer lugar nos bajamos los fuentes de <http://fabrice.bellard.free.fr/qemu/download.html>. Necesitamos el código fuente de qemu (qemu-0.7.2.tar.gz) y el código fuente del acelerador kqemu (kqemu-0.7.2.tar.gz).

Suponemos que los descargamos ambos en el directorio /download.

Lo primero es descomprimirlos:

```
# cd /download
# tar -zxf qemu-0.7.2.tar.gz
# cd qemu-0.7.2
# tar -zxf ../kqemu-0.7.2.tar.gz
```

Se puede apreciar como kqemu lo hemos descomprimido dentro de qemu.

Ahora configuramos la compilación. Para esto tenemos que tener en cuenta que qemu todavía no está preparado para ser compilado con una versión 4.x de gcc.

Para comprobar la versión de gcc que tenemos podemos hacer gcc --version o ls -l /usr/bin/gcc y ver a donde apunta el enlace. Si vemos que tenemos una versión de gcc 4.x (como es mi caso) deberemos indicar al configurador que use una versión 3.x. Para ello utilizamos la opción --cc=gcc-3.3, siendo gcc-3.3 el compilador de la versión 3.x que tengamos instalado en /usr/bin.

El comando quedaría:

```
./configure --enable-adlib --cc=gcc-3.3
```

- Con la opción --enable-adlib conseguimos poder habilitar manualmente la emulación del sonido (SB16) con la opción -enable-audio de qemu.

- La instalación por defecto de qemu es en /usr/local, si queremos instalarlo en otro sitio podemos añadir la opción de configuración --prefix (por ejemplo --prefix=/usr/local/qemu).

En la salida del comando configure tenemos que asegurarnos que tenemos activo el soporte SDL y kqemu. Si no tenemos SDL puede ser porque no tenemos instaladas las librerías de desarrollo, para instalarlas podemos hacer (después tendremos que volver a ejecutar el comando configure para comprobar que ya tenemos soporte para SDL):

```
# apt-get -u install libSDL1.2-dev
```

Esto nos instalará las librerías de desarrollo de SDL y sus dependencias:

```
libaa1-dev (1.4p5-29)
libartsc0-dev (1.4.2-4)
libasound2-dev (1.0.9-3)
libaudiofile-dev (0.2.6-6)
libesd0-dev (0.2.36-1)
libglu1-xorg-dev (6.8.2.dfsg.1-7)
libice-dev (6.8.2.dfsg.1-7)
libSDL1.2-dev (1.2.7+1.2.8cvs20041007-5.3)
libslang2-dev (2.0.4-7)
libsm-dev (6.8.2.dfsg.1-7)
libsvga1-dev (1:1.4.3-22)
libx11-dev (6.8.2.dfsg.1-7)
libxext-dev (6.8.2.dfsg.1-7)
libxi-dev (6.8.2.dfsg.1-7)
libxkbfile-dev (6.8.2.dfsg.1-7)
libxt-dev (6.8.2.dfsg.1-7)
x-dev (6.8.2.dfsg.1-7)
xlibmesa-g1-dev (6.8.2.dfsg.1-7)
```

Ahora es el momento de compilar qemu y kqemu.

Aquí tenemos que tener en cuenta que para que funcione la aceleración con kqemu necesitamos tener los fuentes y la compilación de nuestro kernel en /usr/src/linux, ya que al compilar kqemu se va a generar un módulo (como un driver) para el kernel del sistema.

Los comandos para compilar e instalar son:

```
# make
# make install
```

y todo debería funcionar correctamente.

4. Configuración de la aceleración de qemu con kqemu

Sólo si pasaste por el punto 3.1 !!!

Si hemos instalado qemu con el módulo de aceleración kqemu, para que este funcione tenemos que levantar el módulo del kernel kqemu. Para esto podemos hacer:

```
# modprobe kqemu major=0
```

- La opción de major=0 es necesaria para que udev (el gestor de dispositivos) cree el dispositivo /dev/kqemu. Este dispositivo también es necesario para poder ejecutar qemu con la aceleración.

Si queremos que el módulo del kernel se cargue automáticamente en el arranque del sistema, podemos añadir al fichero /etc/modules la siguiente línea:

```
# Para la aceleración de qemu con kqemu
kqemu major=0
```

También es recomendable que, para que cualquier usuario pueda usar el dispositivo /dev/kqemu, modifiquemos los permisos con los que udev crea este dispositivo. Esto lo conseguimos añadiendo al fichero /etc/udev/permissions.rules la siguiente línea:

```
# Permisos para /dev/kqemu
KERNEL=="kqemu", MODE="0666"
```

5. Creación de una imagen de disco

Ya queda poco para poder instalar Windows XP, pero primero tenemos que crear una imagen de disco. La imagen de disco es un fichero que representará el disco duro donde vamos a instalar Windows XP (o cualquier otro sistema operativo).

Para crear este fichero usamos:

```
$ qemu-img create qemu-winxp.img 10G
```

Es decir estamos creando el fichero con nombre qemu-winxp.img y con un tamaño de 10 GigaBytes (será el tamaño de disco duro que verá el Windows XP).

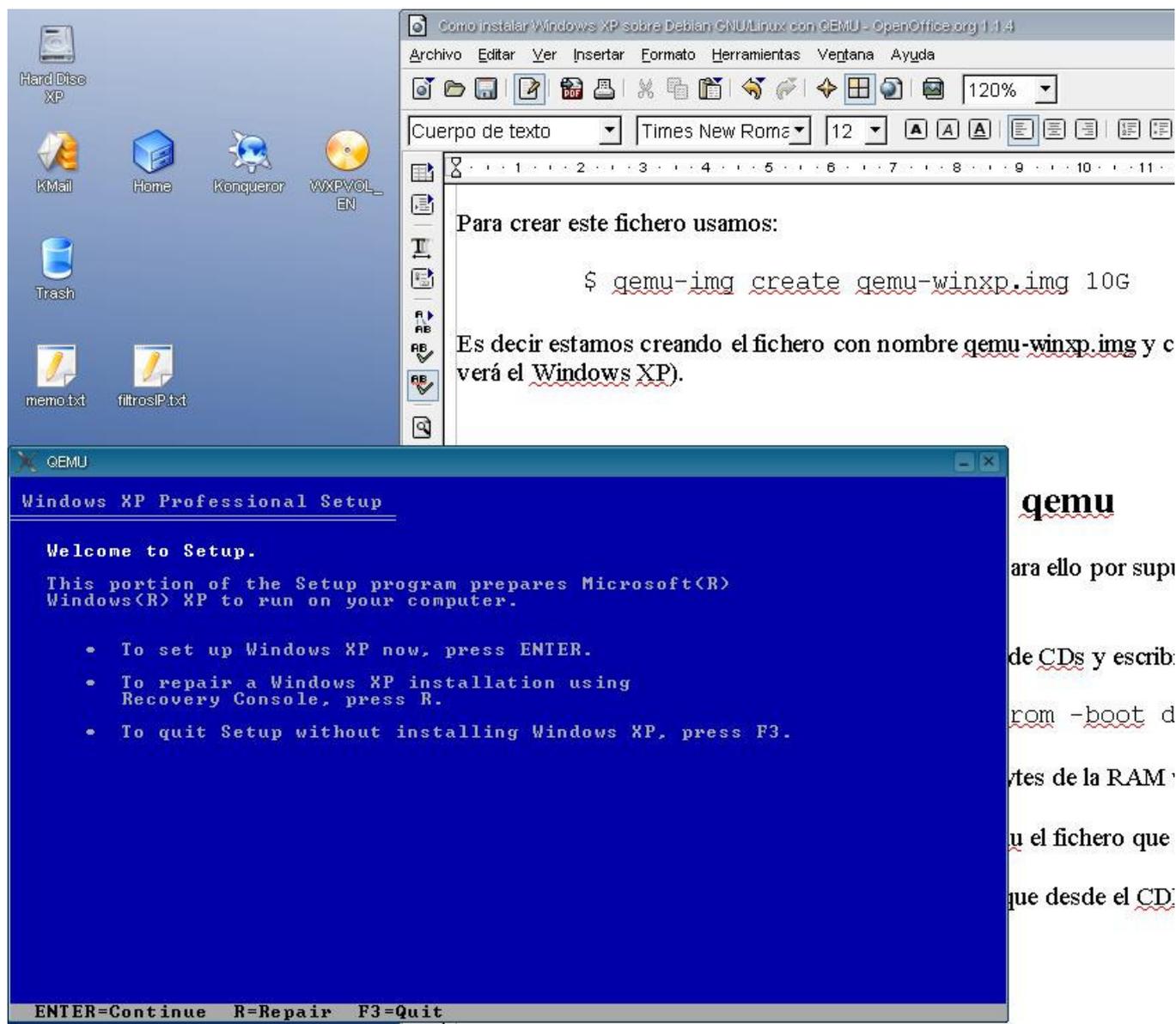
6. Instalación de Windows XP en qemu

Por fin hemos llegado al momento de la instalación. Para ello por supuesto necesitamos un CD de Windows XP con su correspondiente licencia.

Pondremos el CD de Windows XP en nuestro lector de CDs y escribimos el comando:

```
$ qemu -m 512 -cdrom /dev/cdrom -boot d qemu-winxp.img
```

- Con `-m 512` estamos fijando el tamaño en Mega Bytes de la RAM virtual (por supuesto el tamaño de la RAM virtual siempre tiene que ser menor que el tamaño de la RAM real que tiene nuestra máquina).
- Con `-cdrom /dev/cdrom` estamos indicando a qemu el fichero que representa el CDROM IDE que verá la máquina virtual.
- Con `-boot d` estamos indicando a qemu que arranque desde el CDROM (el que hemos indicado con el parámetro anterior).



Ahora continuamos con todo el proceso de instalación de Windows XP.

Cuando estamos dentro de la ventana de qemu, para conseguir que el ratón y el teclado vuelvan a afectar a nuestras X y no a la emulación, tenemos que pulsar Ctrl + Alt.

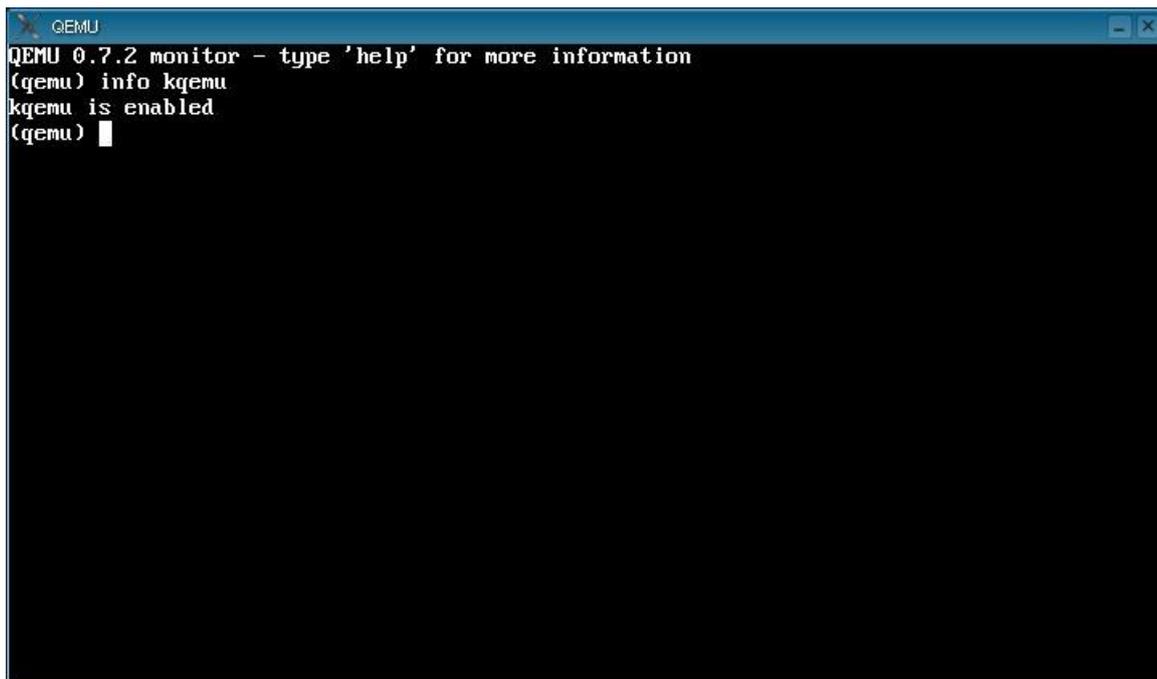
7. Comprobación de la aceleración con kqemu

Sólo si pasaste por el punto 3.1 y 4 !!!

Podemos comprobar si la aceleración está funcionando correctamente desde la consola de qemu.

Para acceder a la consola pulsamos Ctrl + Alt + 2.

En la consola podemos escribir ciertos comandos para interactuar con la máquina virtual. Ahora pondremos 'info kqemu' y nos debe salir un mensaje indicando que está habilitada.



```

QEMU 0.7.2 monitor - type 'help' for more information
(qemu) info kqemu
kqemu is enabled
(qemu) █
  
```

Para volver a la emulación pulsamos Ctrl + Alt + 1.

Si queremos deshabilitar el uso de la aceleración, por ejemplo para comprobar si un error se debe a esta, podemos arrancar qemu con el parámetro -no-kqemu.

8. Configuración de red

Para que nuestra máquina virtual tenga acceso a Internet es tan simple como arrancar qemu con:

```
$ qemu -m 512 -net user -smb /download qemu-winxp.img
```

Con esto obtenemos la siguiente configuración de la red virtual:

```

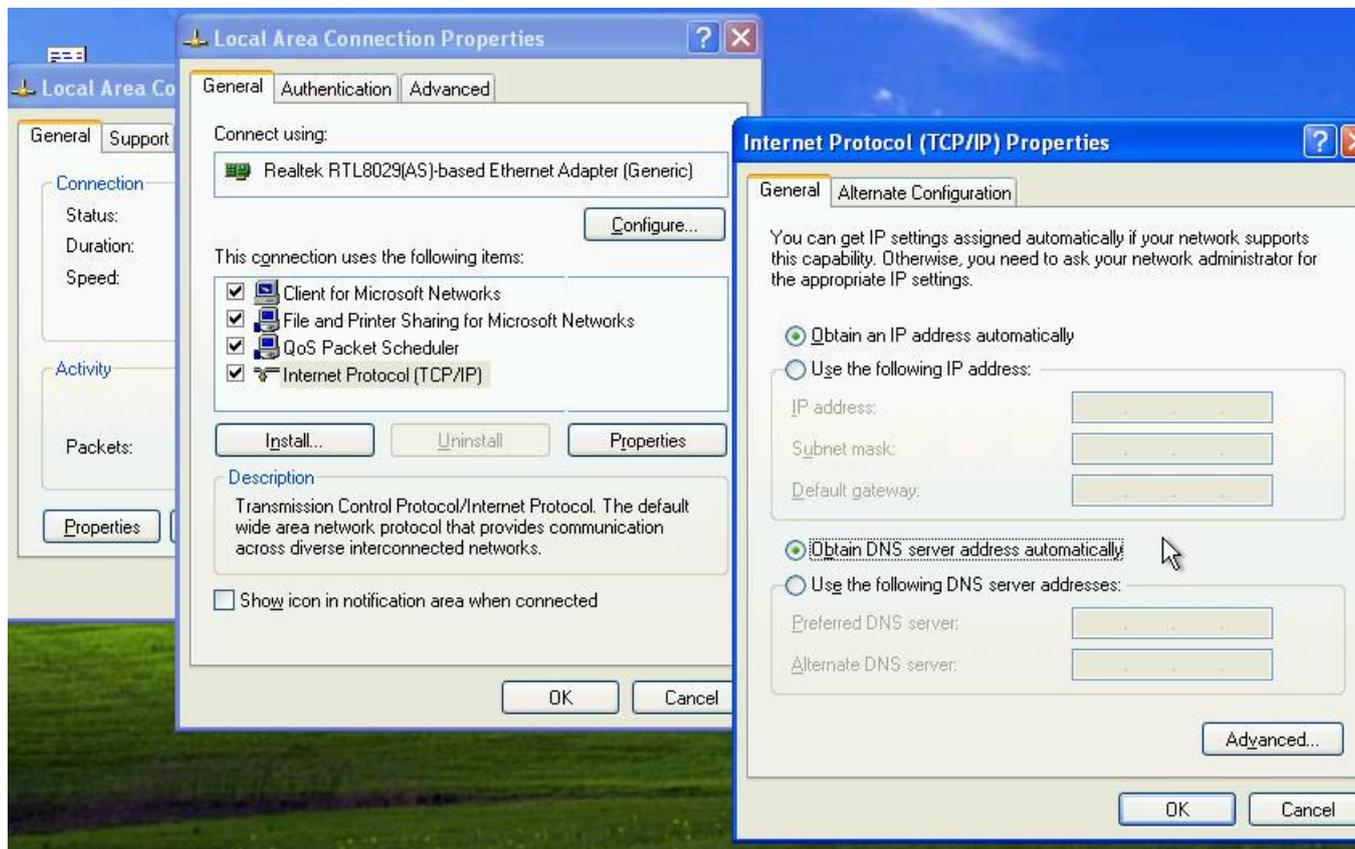
QEMU Virtual Machine <-----> Firewall/DHCP server <-----> Internet
(10.0.2.x)          |          (10.0.2.2)
                    |
                    |-----> DNS server (10.0.2.3)
                    |
                    |-----> SMB server (10.0.2.4)
  
```

Es decir nuestra máquina virtual verá un servidor DHCP en la dirección 10.0.2.2, un servidor de DNS en la dirección 10.0.2.3, y podrá acceder al directorio /download (el directorio que pusimos en la opción -smb) a través de la dirección 10.0.2.4.

En nuestro Windows XP tan sólo tenemos que activar la opción para que se obtenga la dirección IP automáticamente, es decir, que la dirección IP nos la otorgue el servidor DHCP. Para esto, dentro del Windows XP hacemos:

```

Inicio --> Panel de Control --> Conexiones de Red --> Conexión de Área Local --> Propiedades --> Protocolo Internet
(TCP/IP) --> Obtener dirección IP automáticamente y Obtener nombre de servidor DNS automáticamente
  
```



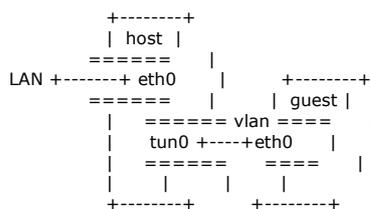
Una vez hecho esto podemos abrir el Explorer dentro del XP y acceder a Internet.

Esta configuración de red es muy sencilla pero el problema que tiene es que no podemos interactuar con la máquina virtual como si fuera otra máquina en nuestra red.

8.1. Configuración avanzada de red

En este punto vamos a ver como podemos configurar la la red virtual para que veamos la máquina virtual como otra máquina dentro de nuestra red. Esto puede ser especialmente útil si queremos instalar dentro del XP algún tipo de servicio, por ejemplo un servidor SQL Server o un servicio Web en .NET.

La configuración de red que vamos a obtener es:



Donde tun0 tendrá la dirección 172.20.0.1 y eth0 en la máquina virtual (guest) tendrá la dirección 172.20.0.2.

Lo primero que necesitamos es tener activo el módulo TUN en nuestro kernel. Para comprobar si tenemos este módulo podemos comprobar la existencia del fichero tun.ko, haciendo:

```
# ls /lib/modules/2.6.12<nuestra versión del kernel>/kernel/drivers/net/
```

Para activar este módulo al compilar el kernel, lo podemos encontrar en:

```
Device Drivers -> Networking support -> Universal TUN/TAP device driver support
```

También tenemos que tener permisos sobre el dispositivo /dev/net/tun, para poder ejecutar qemu con cualquier usuario que no sea root. Para esto podemos hacer:

```
# chmod a+rw /dev/net/tun
```

Si queremos que los permisos se pongan en el arranque tenemos que tocar en un par de sitios. Primero añadimos en el fichero /etc/udev/permissions.rules la línea:

```
KERNEL=="tun", MODE="0666"
```

Y segundo, en el fichero `/etc/udev/links.conf`, en la línea `'M net/tun c 10 200'` añadimos `'--mode=666'`, de forma que la línea quede:

```
M net/tun c 10 200 --mode=666
```

Ahora necesitamos un pequeño script para levantar el interfaz de red que hará de túnel entre nuestro Linux y la máquina virtual. Por defecto `qemu` busca este script en `/etc/qemu-ifup`. Como queremos que `qemu` se pueda usar por cualquier usuario, lo que vamos a hacer es poner todos los comandos de `root` necesarios para levantar el interfaz en otro scrip, y utilizar `sudo` para ejecutar este segundo script. De esta forma podemos configurar fácilmente `sudo` para permitir a usuarios normales levantar la interfaz de red de la máquina virtual.

Con esta idea creamos el script `/etc/qemu-ifup` con el contenido:

```
#!/bin/sh
sudo -p "Pon la clave de tu usuario: " /etc/qemu-ifup-tun $1
```

Le damos permisos de ejecución para todo el mundo:

```
# chmod a+x /etc/qemu-ifup
```

Creamos el segundo script, lo llamaremos `/etc/qemu-ifup-tun`, y tendrá el siguiente contenido:

```
#!/bin/sh
# Se levanta el interfaz que hace de túnel con la máquina virtual
# y se enruta el tráfico de red para que la máquina virtual pueda acceder
# a Internet a través de eth0.
ifconfig $1 172.20.0.1
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

A este scrip sólo le damos permisos de ejecución para `root`:

```
# chmod u+x /etc/qemu-ifup-tun
```

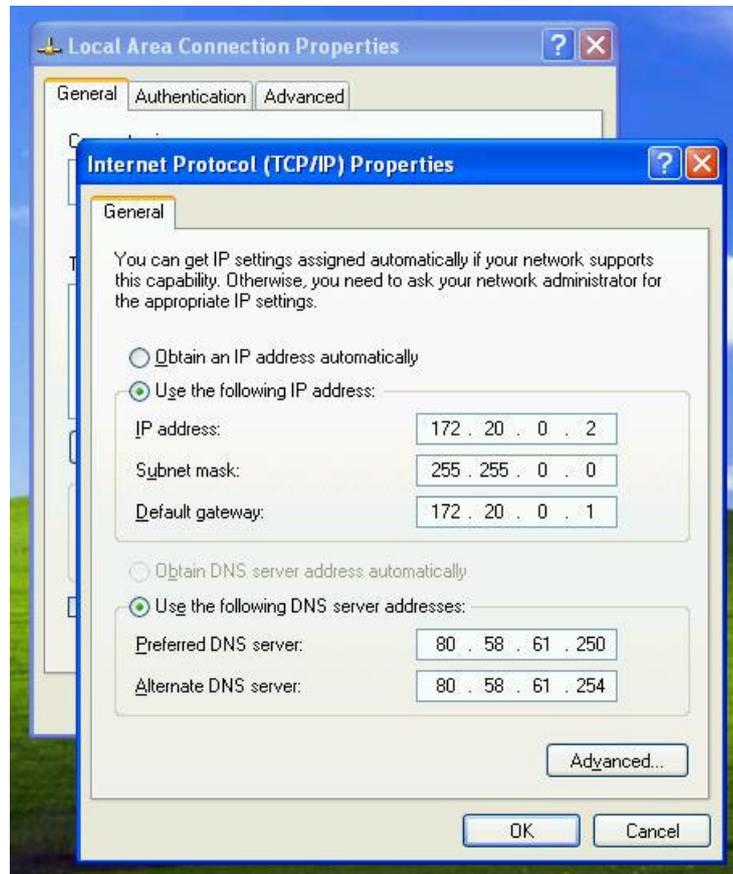
Por último configuramos `sudo` para que los usuarios que queramos puedan ejecutar el script `/etc/qemu-ifup-tun`. Por ejemplo, para que el usuario `alex` pueda ejecutar este scrip, añadimos al fichero `/etc/sudoers` las líneas (conviene editarlo con `visudo`):

```
Cmnd_Alias QEMU = /etc/qemu-ifup-tun
alex ALL= QEMU
```

Ahora para arrancar la imagen podemos usar:

```
$ qemu -m 512 -net nic -net tap qemu-winxp.img
```

El último paso de la configuración avanzada es configurar la conexión de red del XP:



Los valores de DNS son los que utilizáis normalmente, por ejemplo en el `/etc/resolv.conf` o en `/etc/network/interfaces`.

Ahora vuestra máquina virtual con XP puede acceder a Internet, y podéis acceder a ella a través de la IP 172.20.0.1 como si se tratará de cualquier otra máquina. Por ejemplo podéis probar a hacerle ping.

9. Conclusiones

Como hemos visto en este tutorial, los emuladores son herramientas muy potentes. Nos permiten emular otras arquitecturas, o instalar otros sistemas operativos dentro del nuestro.

También nos permite simular máquinas que ofrezcan determinados servicios en nuestra red.

En Autentia (<http://www.autentia.com>), a menudo utilizamos este tipo de herramientas para preparar entornos para demostraciones. De forma que un comercial que quiera hacer una demostración en un cliente no tiene que instalar en su portátil todo el entorno necesario (a veces imposible si la demostración requiere varias máquinas), sino que simplemente se copia el fichero con la imagen y lo ejecuta con el emulador.

Aunque por supuesto la emulación siempre es un poco más lenta que la ejecución real, os puedo asegurar que sólo con la ventaja que nos aporta a la hora de hacer demostraciones y la facilidad de tener disponible un entorno complejo en cualquier máquina tan solo copiando un fichero, ya merece la pena. Además las mejoras de velocidad y herramientas como `kqemu` hacen que esta diferencia sea cada día más pequeña.

`Qemu` también está disponible para Windows, así que también podéis hacer lo contrario, instalar Linux sobre Windows.

10. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software)

Dir. Implantación y Rendimiento

Formador en tecnologías J2EE, ADOO, UML

<mailto:alejandrog@autentia.com>

Autentia Real Business Solutions S.L.

<http://www.autentia.com>

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con



[Autentia S.L.](#) Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto	Descripción
Auto-gestión de dispositivos USB en Debian	En este tutorial y gracias a HAL (Hardware Abstraction Layer) vamos a ver como se puede gestionar de forma automática cualquier dispositivo que tengamos conectado en nuestro ordenador, discos duros, USB, firewire, ...
Crear discos RAM en Windows	Os mostramos como instalar los drivers necesarios para poder montar un disco RAM. Gracias a este mecanismo podemos acceder a una zona de memoria RAM como si fuera un disco duro, autentando el rendimiento en aplicaciones críticas.
Activación de la seguridad en Apache	Alejandro Pérez nos enseña como securizar Apache a través de autenticación básica y certificados de seguridad SSL.
Ejecución de entorno Unix en Windows	Os enseñamos como ejecutar simultaneamente un entorno Windows y Unix en el mismo equipo gracias a CygWin (herramienta gratuita)
Primer vistazo a NetBeans 4	Os mostramos como construir con pocos click de ratón un esqueleto básico de una aplicación Web, al mismo tiempo que comprobamos la estabilidad de la primera beta de la versión 4 de NetBeans
Cliente y Servidor DCOM con MS Visual C++	Os mostramos como construir un servidor y cliente de Automatización OLE con las facilidades que proporciona el entorno de desarrollo Microsoft Visual C++
Informar a Debian de la instalación de Java	En este tutorial os mostramos como informar a Debian GNU/Linux de nuestra Máquina Virtual Java, no instalada desde paquetes Debian.
Nombrar dispositivos USB en Debian	En este tutorial vamos a ver como podemos asignar nombres a los dispositivos USB en Debian, independientemente del nombre que le de el kernel.
Instalación LAMP en Debian	Alejandro Perez nos enseña como instalar nuestro entorno Apache, MySQL y php en Debian
Gestión de Errores con Bugzilla	Alejandro Pérez nos enseña como instalar (en Debian) y utilizar Bugzilla, una herramienta gratuita de gestión de errores.

Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporem alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)

