Utilización y Administración avanzada de sistemas GNU/Linux y aplicaciones Software Libre para estudiantes universitarios

Virtualización y redes en GNU/Linux

José María Peribáñez

Utilización y Administración avanzada de sistemas GNU/Linux y aplicaciones Software Libre para estudiantes universitarios: Virtualización y redes en GNU/Linux

por José María Peribáñez

Historial de revisiones
Revisión 1.0 10-04-2007 Revisado por: José María Peribañez

Tabla de contenidos

Capítulo 1. Virtualización y redes

1.1. Virtualización en GNU/Linux

Mediante la virtualización en una sola máquina es posible tener varios sistemas, como si en lugar de un PC tuviéramos varios. Así, un ISP que ofrezca servidores dedicados puede ofrecer varios servidores virtuales sobre una máquina: cada uno funcionará como un servidor independiente con su propia IP, se podrán instalar todo tipo de servicios (DNS, mail, Jabber, web, PostgreSQL) y reiniciar y administrar independientemente.

Al sistema sobre el que se ejecuta el virtualizador se le llama host; a veces también se habla de domain 0, porque cada sistema que se ejecuta se le considera un dominio. A cada uno de los sistemas que se ejecutan sobre el virtualizador se les denomina guest. En algunos casos no hay sistema host porque el virtualizador se ejecuta directamente sobre el hardware.

Hay hardware como los mainframes de IBM, que soportan virtualización. Los Pcs no soportan virtualización: hay instrucciones de modo protegido que impiden ejecutar dos sistemas operativos simultáneamente, pero mediante software se pueden suplir (con bastante complejidad) las carencias de hardware. Así mismo mediante sofware se emulan componentes del hardware como discos, tarjeta gráfica, de red y de sonido.

Además de la virtualización de la CPU, está la del resto del hardware: tarjeta de red, discos, tarjeta gráfica... Para esta parte la mayoría de los proyectos libres toman código de Qemu.

Hay varios métodos de implementar la virtualiación del procesador:

1. Emuladores totales: emulan totalmente el hardware, incluyendo el procesador. Es el caso de Booch. Su rendimiento es muy pobre.
2. Emuladores con compilación JIT: es el caso de Qemu, cuando se ejecuta sin el módulo kqemu. El código necesita "compilarse" para la máquina virtual, si bien como los compiladores JIT de Java se hace sólo la primera vez que se ejecuta el código, luego ya está compilado. Mucho más rápidos que los emuladores totales, pero más lentos que el resto de soluciones. Se puede ejecutar como usuario normal sin instalar nada como root. Se pueden emular procesadores distintos.
3. Virtualizadores completos: es el caso de Vmware (software privativo), Qemu con el módulo de aceleración Kqemu (antes privativo ahora libre) y Virtual Box (software con una versión libre y otra más funcional privativa). Se basan en un VMM (Virtual Machine Monitor, también conocido como hypervisor) que mediante complicadas técnicas (como traps y análisis del código antes de su ejecución) detecta en tiempo de ejecución el código que no puede ejecutarse directamente porque

afectaría a todo el sistema, modificando dinámicamente las instrucciones conflictivas. El rendimiento varía mucho según lo avanzado que sea el VMM: Vmware tiene varias patentes. Estos tres programas permiten instalar un sistema operativo sobre un virtualizador del mismo modo que sobre un PC: la ventana del virtualizador asemeja el monitor de un PC, podemos entrar en la BIOS, reiniciar el ordenador...

4. Paravirtualizadores: es el caso de Xen y UML (User mode Linux). En lugar de tener que detectar un VMM los casos conflictivos en tiempo de ejecución, se modifica el código fuente de los sistemas operativos para evitar esos casos conflictivos. En lugar de el VMM tener que analizar el código, es el código quien invoca al VMM cuando sea necesario. Esta técnica simplifica muchísimo el VMM y ofrece muy buen rendimiento, aunque en el caso concreto de UML el rendimiento es mediocre. La pega es que haya que parchear el sistema operativo, sobre todo para poder ejecutar Windows. Xen parcheó un Windows XP en un programa de investigación que permitía acceso al código fuente de Microsoft, pero ese tipo de licencias no permitía distribuir el resultado. UML también se puede considerar dentro de esta categoría, pues es una modificación del kernel de Linux para que pueda ejecutarse dentro de otro Linux. Xen no emula una tarjeta gráfica SVGA "completa" como Qemu, Vmware o VirtualBox, pero utiliza VNC que para el sistema guest se ve como una tarjeta VGA.

   Vmware y VirtualBox no son paravirtualizadores, pero utilizan esta técnica para virtualizar la E/S en los drivers especiales que se ejecutan en el sistema guest (por ejemplo el driver de red y el de la tarjeta gráfica: se comunican con el hipervisor en lugar de ser drivers normales sobre el hardware emulado). Lo mismo planea hacer KVM.

5. Virtualizadores apoyados en el hardware (un tanto pretenciosamente llamados nativos): los nuevos procesadores de Intel (los Core Duo y la mayoría, pero no todos, ver la wikipedia, de los Core Duo2 añaden las extensiones VT) y los más recientes de AMD (a partir de stepping F; las extensiones se llaman SVM) cuentan con nuevas instrucciones que permiten la virtualización de la CPU. De este modo ya no es necesario ni un complicado VMM ni parchear el sistema operativo, si bien sigue siendo necesario virtualizar otros dispositivos y emular discos, tarjetas gráficas, de red... Ejemplos de estas soluciones son KVM (integrado en el kernel desde la versión 2.6.20, utiliza qemu para la virtualización del resto del hardware) y Virtual Iron (basado en Xen, pero ya no es paravirtualizador; en cualquier caso es una solución propietaria, con algo de código bajo GPL). Además Xen soporta también estas instrucciones, como método para poder ejecutar Windows o simplemente sistemas sin paravirtualizar. Los virtualizadores apoyados en el hardware son más lentos que los paravirtualizadores e incluso que los virtualizadores completos, al menos que los que tienen un VMM avanzado. VirtualBox no usa por defecto estas instrucciones, aunque las soporte, por este motivo. Lo mismo ocurre con Vmware, aunque sí lo utiliza para poder utilizar como host un sistema operativo de 64bits (VirtualBox actualmente no permite esta configuración).
6. Virtualización a nivel de sistema operativo (OS level Virtualization): no permite ejecutar dos sistemas operativos simultáneamente, sino servidores privados virtuales (SVP) dentro de un único servidor, es decir, es un único kernel, pero que permite aislar (isolate) los servidores. Cada servidor tendrá su propia red, espacio de disco, de memoria, se podrá reiniciar.. así mismo tendrá limitación de uso de CPU con el fin de evitar que un servidor virtual esquilme recursos de los otros. También esta tecnología se denomina como Jail, pues es extender el concepto de chroot. Tantao Linux VServer como Virtuozzo lo vienen ofreciendo proveedores de hosting desde hace años. Estas son las

dos soluciones libres más destacadas:

a. Linux VServer: http://linux-vserver.org (http://linux-vserver.org/) (no confundir con linux virtual server, que es sobre clusters). Software libre. Una limitación es que no admite usar iptables dentro de cada SVP, sino dentro del host. Tampoco admite migración de procesos. Así mismo está más limitado en lo que se virtualiza, por ejemplo no se virtualiza nada de /proc). Por otro lado la distribución host parece menos restringida que en OpenVZ, que como host sólo admite Fedora, algunas versiones de CentOS y algunas de RHEL. Hay varias distribuciones que funcionan como guest directamente (las instalamos en su partición y luego las usamos, con sus programas y librerías, pero no obviamente con su kernel) aunque otras como Gentoo no.

b. OpenVZ (http://openvz.org (http://openvz.org/)). Virtuozzo es un virtualizador bajo una licencia privativa; OpenVZ es el producto de la misma compañía que es software libre y que es la base de Virtuozzo. Un hecho positivo es que OpenVZ no es un conjunto de código GPL difícil de integrar y sin soporte como ocurre con otros productos en los que hay versión GPL y privativa: también venden soporte para OpenVZ. Admite distintas distribuciones como host, a través de templates, que son repositorios para bajar los paquetes necesarios para esa distribución. Hosting con OpenVZ: http://www.vpslink.com/vps-hosting/. En algún caso es más caro con OpenVZ que con Virtuozzo, por ser menos maduro y requerir más recursos...

Entre las posibilidades más avanzadas de algunos virtualizadores está el migrar en caliente con una indisponibilidad inapreciable un dominio de un servidor a otro. Dentro de los productos libres lo permiten Xen, KVM y OpenVZ. Vmware lo permite sólo en su servidor de pago (EXS). En Vmware Server (antiguo Vmware GSX), ahora gratis (que no libre) no es posible. Para saber más sobre este tema: http://kvm.qumranet.com/kvmwiki/Migration

Un debate interesante es el de los virtualizadores que se ejecutan "bare metal", es decir, directamente sobre el hardware, en lugar de sobre un sistema operativo "host". Vmware EXS sigue esta fórmula de forma pura, de tal modo que sólo funciona con determinado hardware. En el caso de Xen, se ejecuta sobre un host, pero asume parte de las funciones de un sistema operativo, al tener código por ejemplo para planificar la CPU. En cambio KVM utiliza todo lo que aporta Linux, lo que para muchos desarrolladores es la opción preferible pues difícilmente una empresa que desarrolla un producto de virtualización tendrá más experiencia en elementos de sistemas operativos como un planificador, que el grupo de personas que llevan desarrollando estos componentes desde hace años en un sistema tan extendido como el kernel Linux.

Software interesante:

http://virt-manager.et.redhat.com/ interfaz gráfica sobre todo para Xen, pero también soporta

KVM y Qemu. Incluye parte para gestionar redes.

Información muy completa (navegar por los enlaces del marco de la izquierda):

http://virt.kernelnewbies.org/TechOverview

1.1.1. VMWare y la red

En VMWare a la hora de crear un dispositivo de red se puede elegir entre:

1. NAT: en el host aparecerá como la interfaz vmnet8, pero no podremos hacer nada con esta interfaz, por ejemplo usar Iptables para restringir las Ips que se pueden alcanzar desde el sistema que se ejecuta dentro de Vmware. El motivo es que no se usa realmente el NAT del núcleo sino un demonio, vmnet-natd, por lo que los paquetes no pasan realmente por vmnet8. Si se quiere abrir puertos, basta con editar el fichero /etc/vmware/vmnet8/nat.conf
2. Bridge: también se implementa utilizando un demonio privativo, en este caso vmnet-bridge, en lugar de utilizar el soporte del núcleo, por lo que no se puede restringir la red con iptables sobre la interfaz vmnet2. Como en el caso de vmnet8, realmente el tráfico no pasa por esta interfaz.
3. Host only: aparentemente este modo es el más limitado. Pues bien, en realidad es el más flexible, pues por la interfaz de red que crea, vmnet1, sí que pasan todos los paquetes. De este modo podemos utilizar esta interfaz para hacer NAT a través de iptables, o crear un bridge con brctl. Al usar iptables, podemos restringir el tráfico como con cualquier otra unidad de red.

En Qemu hay dos formas de utilizar la red. Por defecto se usa -net socket, que sería el equivalente al modo NAT de Vmware. Mediante la opción -redir se pueden abrir puertos de servidor. Una diferencia interesante sobre Vmware es que esta solución se implementa enteramente en espacio de usuario, por lo que no se crea interfaz de red ni se precisa cargar ningún módulo del kernel, lo que es bueno porque en el caso de Vmware es un módulo privativo que activa la marca "tained" del kérnel, con lo que perdemos toda opción de soporte. Además así no hay que tener privilegios de superusuario para instalar Qemu (Vmware no requiere privilegios para ejecutarse, pero sí hace falta para insertar el módulo en el kernel).

Las interfaces de red vmnet1, vmnet2, vment8, se crean al ejecutar vmware-config, en la parte de configuración de red. Una posibilidad interesante si vamos a ejecutar varios sistemas simultáneamente o si queremos que un sistema tenga más de una interfaz es crear más de un dispositivo de red para "host

only", de modo que aparte de vmnet1 haya otros. De otro modo todas las máquinas virtuales estarán conectadas a la misma red, la de vmnet1. Así, una máquina podrá tener la IP 192.168.152.128, otra la 192.168.152.129 y el host la 192.168.152.1; las dos máquinas virtuales se verán la una a la otra y podrán comunicarse, aunque eso sí, con un sniffer una máquina no verá el tráfico de los otros nodos.

La red vmnet1 que crea Vmware es de máscara de red 255.255.255.0; ejecuta un servidor DHCP automáticamente, cuya configuración y arrendamientos pueden verse en /etc/vmware/vmnet1. Pero nada impide que podamos cambiar esta configuración, pues se emula a una interfaz ethernet. Por ejemplo un nodo puede cambiar su IP a otra de la red o incluso crear un alias y usar una red distinta, tanto en el host como en las máquinas virtuales.

Si usamos vmnet1 para hacer NAT utilizando iptables, hay que tener en cuenta que habrá que configurar la red de la máquina virtual para añadirle una ruta por defecto y la configuración del DNS y que el firewall deberá permitir que llegue al servidor DNS. Para pasar la configuración de DNS y ruta por defecto podemos usar el servidor DHCP de Vmware: "option routers" y "option domain-name-servers".

1.1.2. Qemu y TUN/TAP

Un inconveniente de -net socket es que como ocurre con los modos NAT y Bridge de Vmware los paquetes no pasan por ninguna interfaz de red, por lo que no se puede utilizar iptables para restringir la red.

La solución está en el uso del soporte de TUN/TAP del kernel. Consiste en que una aplicación puede abrir el fichero de dispositivo /dev/net/tun y con eso crear una nueva interfaz de red (por defecto tun0). Todo lo que la aplicación escriba en ese dispositivo se recibirá en la interfaz de red recién creada; de igual modo todo lo que llegue a esa interfaz de red (por ejemplo a través de enrutamiento) lo leerá la aplicación del fichero de dispositivo.

Los dispositivos TUN operan a nivel IP y son punto a punto. Los dispositivos TAP operan a nivel 2 y son multipunto, como las interfaces eth*. Un dispositivo tap0 y un dispositivo vmnet1 vienen a funcionar de forma muy similar y a nivel de ejemplos de configuración con iptables o brctl donde aparezca un tap0 podría aparecer un vmnet1 y viceversa. Por lo general se usa tun0 en lugar de tap0; para la mayoría de los usos son equivalentes por lo que es mucho más habitual utilizar tun0 que resulta más sencillo y directo.

Normalmente un dispositivo tun/tap sólo existe mientras el programa no cierra el fichero /dev/net/tun. Esto a veces es problemático, especialmente porque para crear un dispositivo TUN/TAP hacen falta privilegios de superusuario. Afortunadamente, con root se puede abrir un dispositivo en modo persistente para un determinado usuario, de modo que luego un programa ejecutado por ese usario sin privilegios podrá abrir ese dispositivo tun/tap creado para él por el root. Este se puede hacer con el programa tunctl, que forma parte del paquete uml-utilities.

¿Por qué no usa Vmware TUN/TAP en lugar de vmnet1? quizás por unicidad entre plataformas, o por diferencia de implementación; es posible que vmnet1 también permita driver de red de la máquina virtual directamente en espacio del kernel.

TUN/TAP es la solución utilizada también por otros virtualizadores, como Virtual Box. Muy interesantes los documentos sobre red avanzada en la web de Virtual Box:
http://www.virtualbox.org/wiki/User_HOWTOS.

1.1.3. Ejemplos

Permitir acceso sólo a red local 192.168.10.0 y además excluir el nodo 9 de esa red.

```
iptables -A FORWARD -i vmnet1 --destination 192.168.10.9 -j REJECT

iptables -A FORWARD -i vmnet1 --destination 192.168.10.0/24 -j ACCEPT

iptables -A FORWARD -i vmnet1 -j REJECT

iptables -t nat -A POSTROUTING -j MASQUERADE -o eth0

echo 1 > /proc/sys/net/ipv4/conf/eth0/forwarding

echo 1 > /proc/sys/net/ipv4/conf/vmnet1/forwarding
```

Crear un bridge, pero prohibiendo el acceso a la IP 157.88.10.20. Se puede filtrar con ebtables o con iptables.

Ojo, muchas tarjetas wireless no soportan bridge. Un AP generalmente sí.

ifconfig eth0 0.0.0.0

ifconfig vmnet1 0.0.0.0

brctl addbr puente

brctl addbr puente

brctl addif eth0

brctl addif vmnet1

ifconfig puente 192.168.10.1

iptables -O FORWARD -m physdev --physdev-in vmnet1 --destination 157.88.10.20 -j REJECT

1.1.4. Instalar QEMU

Para compilar kqemu no hace falta versión vieja de compilador gcc, ni las fuentes de qemu. Es bastante rápido:

./configure

make

sudo make install

modprobe kqemu

Por defecto, necesitamos permisos de root para leer /dev/kqemu

1. creamos grupo qemu: sudo addgroup qemu
2. añadimos nuestro usuario (jomar) al grupo: sudo gpasswd -a jomar qemu
3. configuramos udev para que cree fichero de dispositivo con permisos para grupo qemu. Para ello editamos fichero /etc/udev/rules.d/60-kqemu.rules con este contenido: KERNEL=="kqemu", NAME="%k", MODE="0666", GROUP="qemu", RUN="/root/prueba.sh"
4. hacemos lo propio con el fichero /dev/net/tun. A partir del kernel 2.6.18 no pasa nada por dar permiso para todo el mundo, pues nadie sin privilegios puede crear una nueva interfaz si no se ha creado antes por el root para ese usuario. Esto se puede hacer con la herramienta tunctl, que forma parte del paquete uml-utilities. También se puede usar el programa que adjuntamos más adelante, cambiando el tipo de dispositivo de tun a tap. Para crear el dispositivo con tunctl se usa tunctl -u jomar -t tap0; para borrarlo tunctl -d tap0.

Ejemplos:

./qemu -net nic -net tap,script=no ~/linux-0.2.img -net nic,vlan=1 -net socket,vlan=1,listen=:8081 centos.qcow2

ifconfig tap0 up

ifconfig eth0

brctl addif puente tap0

ifconfig eth0 0.0.0.0

brctl addif puente eth0

ifconfig eth0 192.168.15.45

./qemu -net nic -net tap,script=no ~/centos.qcow2 -no-kqemu

1.1.5. Crear una imagen con Qemu

qemu-img create -f qcow2 centos.qcow2 3G

Con esta orden se crea una imagen de un disco de 3GB; en realidad con GNU/Linux este fichero no ocupa 3GB; el espacio sin usar del fichero no ocupa espacio en el disco.

Para arrancar del CD de instalación podemos ejecutar:

qemu -kernel-kqemu -cdrom /home/jomar/centos1of6.iso -boot d -m 512 centos.qcow2

La opción -kernel-kqemu es para obtener la máxima aceleración: acelera también el código del espacio del kernel; por defecto sólo se acelera la de usuario. Si se produce algún problema, reintentaremos sin esta opción; si sigue habiendo problemas podemos probar con -no-kqemu a quitar incluso la aceleración de espacio de usuario. Tras la instalación, podemos probar a volver a utilizar aceleración: a veces los posibles problemas sólo se dan durante la instalación, aunque esta situación es más propia de Windows que GNU/Linux.

Con la opción -cdrom le indicamos que el CDROM es en realidad esa ISO; así evitamos el tener que tostar un CD. Con la opción -boot d indicamos que arranque de CD en vez de disco duro. La opción -m 512 establece que la máquina virtual tenga 512 MB de memoria. La cantidad de memoria puede cambiarse de una ejecución a otra, pero hay que tener en cuenta que los instaladores suelen crear una partición de intercambio con el doble de la memoria RAM.

¿Cómo cambiar de CD durante la instalación? Con ctrl-alt-2 (ojo 1, no F1) pasamos al monitor, en el que ejecutamos el comando:

change cdrom /home/jomar/centos2of6.iso

Tras escribir la orden (qemu no indica ni errores ni que la operación se ha realizado con éxito) volvemos a la pantalla de la máquina virtual con ctrl-alt-1.

Una característica interesante de Qemu es que permite crear a partir de un fichero de imagen, otra imagen que sólo contendrá los cambios que se produzcan en la primera, si la primera permanece sin modificar. Esta característica es útil por ejemplo para crear dos imágenes muy parecidas para hacer pruebas de redes entre las dos imágenes, sin que ocupe tanto en el disco.

qemu-img create -b xubuntu.qcow2 xubuntu1.qcow2

qemu-img create -b xubuntu.qcow2 xubuntu2.qcow2

1.1.6. Listados de ejemplos de TUN/TAP

Listado para usar el dispositivo creado para este usuario (en kernels viejos lo crea si no existe): muestra primer paquete que recibe, una salida similar a la de tcpdump -x -i tun0. Para probarlo configuramos con ifconfig el dispositivo tun0, usando pointtopoint para indicar una IP supuestamente destino; podemos hacer un ping a esa dirección y ver lo que recibe el programa.

Listado para crear como persistente el nodo:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <linux/if_tun.h>
#include <linux/if.h>
#include <string.h>

int main() {
   struct ifreq ifr;
   int fd, err;
   char dev[IFNAMSIZ];

   fd = open("/dev/net/tun", O_RDWR) ;
   memset(&ifr, 0, sizeof(ifr));

   /* Flags: escoger entre IFF_TAP o IFF_TUN; opcionalmente
    * añadir IFF_NO_PI
    * IFF_TAP   - Dispositivo TAP: Nivel ethernet (Layer2),
    *             multipunto; útil para hacer bridge o para tráfico no IP
    *             también para Qemu, pues usa TAP
    * IFF_TUN   - Dispositivo TUN: nivel IP (Layer3); punto a
    *             punto; la opción recomendada normalmente si
    *             sólo trafico IP sin bridge
    * IFF_NO_PI - No añadir a paquetes TUN información adicional
    *             antes del paquete.
    */
    ifr.ifr_flags = IFF_TUN|IFF_NO_PI;
    if( *dev )
        strncpy(ifr.ifr_name, "tun0",5);
    err= ioctl(fd, TUNSETIFF, (void *) &ifr);
    if (!(err<0)) {
      // Hacer dispositivo persistente, es decir, que permanece
      // creado aunque se cierre dispositivo; normalmente el
      // dispositivo sólo existe mientras está abierto el
      // descriptor de fichero a /dev/net/tun.
      // Cambiar 1 por 0 para borrar un dispositivo persistente.
      err= ioctl(fd, TUNSETPERSIST, 1);
```

```
      // Poner como dueño del dispositivo al UID 1000; de este
      // modo puede crear dispositivo proceso ejecutado por
      // usuario con este UID en lugar de root, si además tiene
      // permiso de lectura y escritura sobre /dev/net/tun
      // antes de kernel 2.6.18 no era necesario hacer esto, pero
      // ahora ya sí. Con Qemu otra alternativa es hacer un
      // programa lanzador setuid que abra fichero, revoque
      // privilegios y lance Qemu pasándole handler abierto; o que
      // un programa con privilegio de root abra fichero y pase el
      // handler vía un socket UNIX (ver man cmsg)
      if (!(err<0))
          err= ioctl(fd, TUNSETOWNER, 1000);
    }

    if( err<0) {
        perror("falló");
        close(fd);
        return err;
    }
    strcpy(dev, ifr.ifr_name);
    printf("%s\n",ifr.ifr_name);
    return 0;
}
```

Programa para crear un dispositivo TUN/TAP persistente

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <linux/if_tun.h>
#include <linux/if.h>
#include <string.h>

int main() {
   struct ifreq ifr;
   int fd, err;
   char dev[IFNAMSIZ];

   fd = open("/dev/net/tun", O_RDWR) ;
   memset(&ifr, 0, sizeof(ifr));

   /* Flags: escoger entre IFF_TAP o IFF_TUN; opcionalmente
    * añadir IFF_NO_PI
    * IFF_TAP   - Dispositivo TAP: Nivel ethernet (Layer2),
    *             multipunto.
    *             útil para hacer bridge o para tráfico no IP
    *             también para Qemu, pues usa TAP
```

```
    * IFF_TUN   - Dispositivo TUN: nivel IP (Layer3); punto a
    *             punto; la opción recomendada normalmente si
    *             sólo tráfico IP sin bridge
    * IFF_NO_PI - No añadir a paquetes TUN información adicional
    *             antes del paquete.
    */
    ifr.ifr_flags = IFF_TUN|IFF_NO_PI;
    if( *dev )
        strncpy(ifr.ifr_name, "tun0",5);
    err= ioctl(fd, TUNSETIFF, (void *) &ifr);
    if (!(err<0)) {
      // Hacer dispositivo persistente, es decir, que permanece
      // creado aunque se cierre dispositivo; normalmente el
      // dispositivo sólo existe mientras está abierto el
      // descriptor de fichero a /dev/net/tun.
      // Cambiar 1 por 0 para borrar un dispositivo persistente.
      err= ioctl(fd, TUNSETPERSIST, 1);

      // Poner como dueño del dispositivo al UID 1000; de este
      // modo puede crear dispositivo proceso ejecutado por
      // usuario con este UID en lugar de root, si además tiene
      // permiso de lectura y escritura sobre /dev/net/tun
      // antes de kernel 2.6.18 no era necesario hacer esto, pero
      // ahora ya sí. Con Qemu otra alternativa es hacer un
      // programa lanzador setuid que abra fichero, revoque
      // privilegios y lance Qemu pasándole handler abierto; o que
      // un programa con privilegio de root abra fichero y pase el
      // handler vía un socket UNIX (ver man cmsg)
      if (!(err<0))
          err= ioctl(fd, TUNSETOWNER, 1000);
    }

    if( err<0) {
        perror("falló");
        close(fd);
        return err;
    }
    strcpy(dev, ifr.ifr_name);
    printf("%s\n",ifr.ifr_name);
    return 0;
}
```

Apéndice A. GNU Free Documentation License

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of

Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and

3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

GNU FDL Modification Conditions

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is

included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or

rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Sample Invariant Sections list

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

Sample Invariant Sections list

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public

License, to permit their use in free software.