

Objetivo:

El objeto de esta práctica es describir el proceso necesario para arrancar el emulador qemu, dotarlo de conexión de red (virtual) con la máquina local, y montar SENU y GECO de forma que se pueda trabajar con ellos desde cualquier máquina de la red que esté en windows (incluida la emulada)

Eso se ve mejor en **foto1 foto2 y foto3** (adjunta a este documento).

Instalación y manejo básico de qemu

Lo primero:

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

El siguiente paso es “crearle” un disco duro en un fichero. Es decir, toda la información que se debe guardar en la unidad C: del windows, se guardará en un fichero de linux. (Así hacer una copia de seguridad de todo el windows virtual....)

Para ello debemos decidir qué queremos instalar en la máquina de W98 (pocas cosas). Como curiosidad, he instalado Office y funciona, pero...para qué.

Tanto SENU como GECO caben en un disquete, así que el tamaño del disco C: no hace falta que sea muy grande. Necesitamos que entre el sistema operativo, y poco más.

La orden

```
$qemu-mkcow hd.img 600
```

creará un fichero que en linux se verá como hd.img, y que para W98 será un disco duro.

Arranque de la máquina emulada

Ya está, ahora sólo hay que arrancarlo.

Si hacemos

```
$qemu -hda hd.img
```

arrancaremos el qemu, pero como no podrá cargar sistema operativo, pues no podremos hacer nada.

Así que debemos decirle en el arranque más cosas. Por ejemplo, que arranque desde el cdrom en el que tenemos metido el disco de windows98

Con la orden

```
$qemu -hda hd.img -cdrom /dev/cdrom -boot d
```

NOTA podemos hacer que qemu trabaje con un disco duro físico en vez de con el fichero de imagen
Para ello bastaría con hacer

```
$qemu -hda /dev/hdc -cdrom /dev/cdrom -boot d
```

y nos dispondríamos a usar el máster del segundo canal de nuestro equipo como máster del primer canal de la máquina emulada.

A partir de aquí el proceso consiste en instalar W98 como de costumbre, aunque no podremos hacer disco de arranque porque no le hemos dicho que use la disquette.

Si pensamos que vamos a usar la disquette, el comando debería ser

```
$qemu -hda hd.img -cdrom /dev/cdrom -boot d -fda /dev/fd0.
```

Hay que tener cuidado porque, aunque veamos la disquette desde W98, es linux quien realmente la gestiona, por lo que no tenemos la facilidad windows de cambiar el disco y ya está, sino que, de alguna manera, hay que decirle a qemu que desmonte y vuelva a montar la unidad.

En este punto, debemos detenernos en el manejo de la consola de qemu. A través de ella controlamos todo lo que nuestros dedos nos permiten controlar en un puesto de trabajo (encendido, apagado, reinicio, cambio de disquettes y Cds) y más cosas.

Con CTRL+MAYUS sueltas la emulación y puedes ir cualquier parte de guadalinux (deja de funcionar el ratón de W98 y el que funciona es el de guadalinux). Ahí podemos ir a la consola y teclear el comando help

(qemu) help (ojo: (qemu) es el cursor de la línea de comandos de la consola del emulador)

Veremos la lista de comandos disponibles, entre los que está salvar la emulación por donde vaya, cargar otra, expulsar un disco,...)

Configuración de Red.

Suponemos que nuestra máquina local tiene una tarjeta de red con dirección I.P 191.168.0.4 y máscara de sub-red 255.255.255.0. El Gateway es 192.168.0.1

Vamos a dar los pasos necesarios para que cuando la máquina emulada se arranque, pueda incorporarse a la red de trabajo como una máquina más. Para ello tendremos que montar un bridge sobre eth0. Esto es un poco lioso, pero doy los pasos que se deben seguir.

1.º El sistema debe estar compilado con la opción

```
Universal TUN/TAP device driver support
```

y en Networking

```
802.1d Ethernet Bridging
```

A continuación

```
# mkdir /dev/net
# mknod /dev/net/tun c 10 200
# groupadd tuntap
# chmod 0770 /dev/net/tun
# chown root.tuntap /dev/net/tun
```

Debemos añadir el usuario que vaya a correr la máquina emulada al grupo tuntap

Añadimos al final del fichero /etc/modules las líneas

```
tun
```

```
bridge
```

También debemos modificar nuestro fichero /etc/network/interfaces

El mío queda

```
iface br0 inet static
    address 192.168.0.4
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    bridge_ports eth0
    bridge_stp off
    bridge_maxwait 5
    name Tipo de interfaz desconocido

iface lo inet loopback
address 127.0.0.1
netmask 255.0.0.0
broadcast 127.255.255.255
```

```
network 127.0.0.0
```

```
auto lo
```

Cuando se arranca qemu, se ejecuta un script /etc/qemu-ifup

El mío contiene

```
#!/bin/sh
```

```
sudo /sbin/ifconfig $1 192.168.0.18
```

```
sudo /usr/sbin/brctl addif br0 $1
```

Estos comandos son los que construyen el puente.

sudo significa (por la configuración que ahora veremos) que ese comando se ejecuta como root, De esta forma, podemos determinar qué usuarios tendrán acceso a levantar ese puente.

Con el comando visudo (como root) se accede a controlar quienes pueden ejecutar como root qué comandos. Concretamente, he añadido

```
User_Alias USUQEMU = eva #(solo el usuario eva puede levantar el puente)
```

```
Cmnd_Alias CONFIGRED = /sbin/ifconfig, /usr/sbin/brctl
```

Ahora toca el turno arrancar la máquina virtual y configurarla . Para ello: Inicio->configuración->Panel de control->red **(foto4)**

Seleccionamos TCP/IP->Realtek.... y clic en propiedades
Colocamos la I.P 192.168.0.20 con máscara de subred 255.255.255.0

En la pestaña Puerta de enlace colocamos 192.168.0.18 (el mismo que pusimos en /etc/ifup-qemu)

Rellenamos la pestaña identificación con el nombre de nuestra red y nuestro equipo. **(foto5)**

En la máquina linux tenemos que hacer

```
#network-config &
```

y en la pestaña general habilitar el trabajo de red en windows poniendo en Dominio el mismo grupo que en la máquina emulada (W98)

De esta forma se puede conseguir que la máquina emulada vea a la guadalinux (garma4) en mi caso. Y que desde la guadalinux se vea la 192.168.0.20 **(foto6)**

Ahora sólo queda crear los shares correspondientes en /etc/samba/smb.conf, y añadir las carpetas correspondientes como unidades de red en las que luego se instalarán tanto GECO como SENU.

El fichero de smb.conf contiene:

[senu]

```
path = /var/senu
comment = Directorio de senu
writeable = yes
create mode = 0770
directory mode = 0770
valid users = usuario, eva
force user = usuario
force group = senu
guest ok = no
guest only = yes
browseable = yes
```

[geco]

```
path = /var/geco
comment = Directorio de gecko
writeable = yes
create mode = 0770
directory mode = 0770
valid users = usuario, eva
force user = usuario
force group = gecko
guest ok = no
guest only = yes
browseable = yes
```