

# Table of Contents

- QEMU** ..... 3
  - ¿Que es QEMU? ..... 3
  - ¿Que puede virtualizar? ..... 3
  - ¿Que puede emular? ..... 3
  - Instalación ..... 4
  - Hola Mundo de la emulación ..... 4
  - Iniciar un sistema operativo ..... 5
  - Configuración básica de QEMU ..... 6
  - TODO ..... 12
  - Referencias ..... 12



# QEMU

Este artículo pretende dar una pequeña introducción a el software de emulación y virtualización QEMU sobre GNU/Linux, en específico de Debian GNU/Linux.

## ¿Que es QEMU?

QEMU es un software libre que permite emular y virtualizar.

Un emulador permite que su computadora ejecute sistemas operativos y software que han sido escritos para otro tipo de computadora. Por ejemplo podría ejecutar el sistema operativo del teléfono celular en su computadora portátil.

Un virtualizador permite que su computadora ejecute de una manera eficiente sistemas operativos y software que han sido escritos para computadoras similares, por ejemplo podría ejecutar tres versiones de GNU/Linux al mismo tiempo en la computadora de su escritorio.

El sistema operativo que corre sobre su computadora se llama Host o Anfitrión, mientras que el sistema operativo emulado o virtualizado se llama Guest o Huésped. Así pues tenemos que un sistema operativo Host puede emular o virtualizar varios sistemas operativos Guest.

El rendimiento obtenido por la virtualización es mejor que el de la emulación, esto se debe a que el sistema operativo Guest habla que el CPU y demás recursos de su computadora, entonces el Host permite que el Guest se comunique directamente con el CPU y demás recursos. En términos mas técnicos, al virtualizar en un procesador x86\_64 debo utilizar únicamente sistemas operativos Guest que corran en procesadores x86\_64.

Cuando se emula no necesariamente el sistema operativo Guest habla el mismo idioma que el CPU y demás recursos de su computadora, entonces el sistema operativo Host debe traducir las cosas que el Guest dice para que el CPU y los recursos entiendan que desea hacer, esta traducción requiere esfuerzo según el idioma del Guest. En términos mas técnicos, cuando se emula un sistema operativo para ARM, como el que usan los teléfonos, en un procesador x86\_64, como el que usan las computadoras portátiles, el sistema operativo debe traducir de ARM a x86\_64, consumiendo mucho procesamiento para preparar la traducción.

## ¿Que puede virtualizar?

QEMU requiere de un módulo adicional llamado KVM, el cual permite hacer la comunicación entre el Guest y el CPU y los recursos de la computadora, mediante KVM se pueden virtualizar las siguientes plataformas:

- x86
- powerpc
- s390

## ¿Que puede emular?

Con cada versión de QEMU se amplían las plataformas emuladas, hasta la versión 2.6 son:

- x86
- arm
- mips
- powerpc
- sparc
- xtensa
- alpha
- cris
- m68k
- sh4
- lm32
- microblaze
- moxie
- openrisc
- tricore
- unicore

## Instalación

En sistemas operativos basados en Debian GNU/Linux puede instalarlo de la siguiente manera:

```
apt-get install qemu qemu-system
```

## Hola Mundo de la emulación

Para efectos prácticos vamos a emular un sistema operativo x86 en nuestra computadora x86. Lo vamos a hacer mediante la consola de texto, la razón es que no he encontrado un GUI para QEMU que soporte todas las arquitecturas, por lo general se limitan a x86.

Vamos a correr el live-cd de Debian, puede descargarlo de [aquí](#), la imagen debian-live-8.X.0-amd64-standard.iso es la mas pequeña, pero no tiene entorno gráfico, la imagen debian-live-8.X.0-amd64-xfce-desktop.iso es la mas pequeña con entorno gráfico.

```
qemu-system-i386
```

Al ejecutar ese comando una ventana emergente mostrar una computadora intentando iniciar el sistema operativo:

```
QEMU
Boot failed: could not read the boot disk
Booting from DVD/CD...
Boot failed: Could not read from CDROM (code 0003)
Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+git-20150424.a25a16d-1 -- Open Source Network Boot Firmware -- http://
/ipxe.org
Features: DNS HTTP iSCSI NFS TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: 52:54:00:12:34:56 using 82540em on PCI00:03.0 (open)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 52:54:00:12:34:56)..... ok
net0: 10.0.2.15/255.255.255.0 gw 10.0.2.2
net0: fe80::5054:ff:fe12:3456/64
net0: fec0::5054:ff:fe12:3456/64 gw fe80::2
Nothing to boot: No such file or directory (http://ipxe.org/2d03e13b)
No more network devices

No bootable device.
```

De manera predeterminada QEMU selecciona una serie de componentes que permiten iniciar la emulación, en nuestro caso seleccionó una tarjeta de video, una unidad lectora de DVD/CD, un BIOS, una interfaz de red, entre otros.

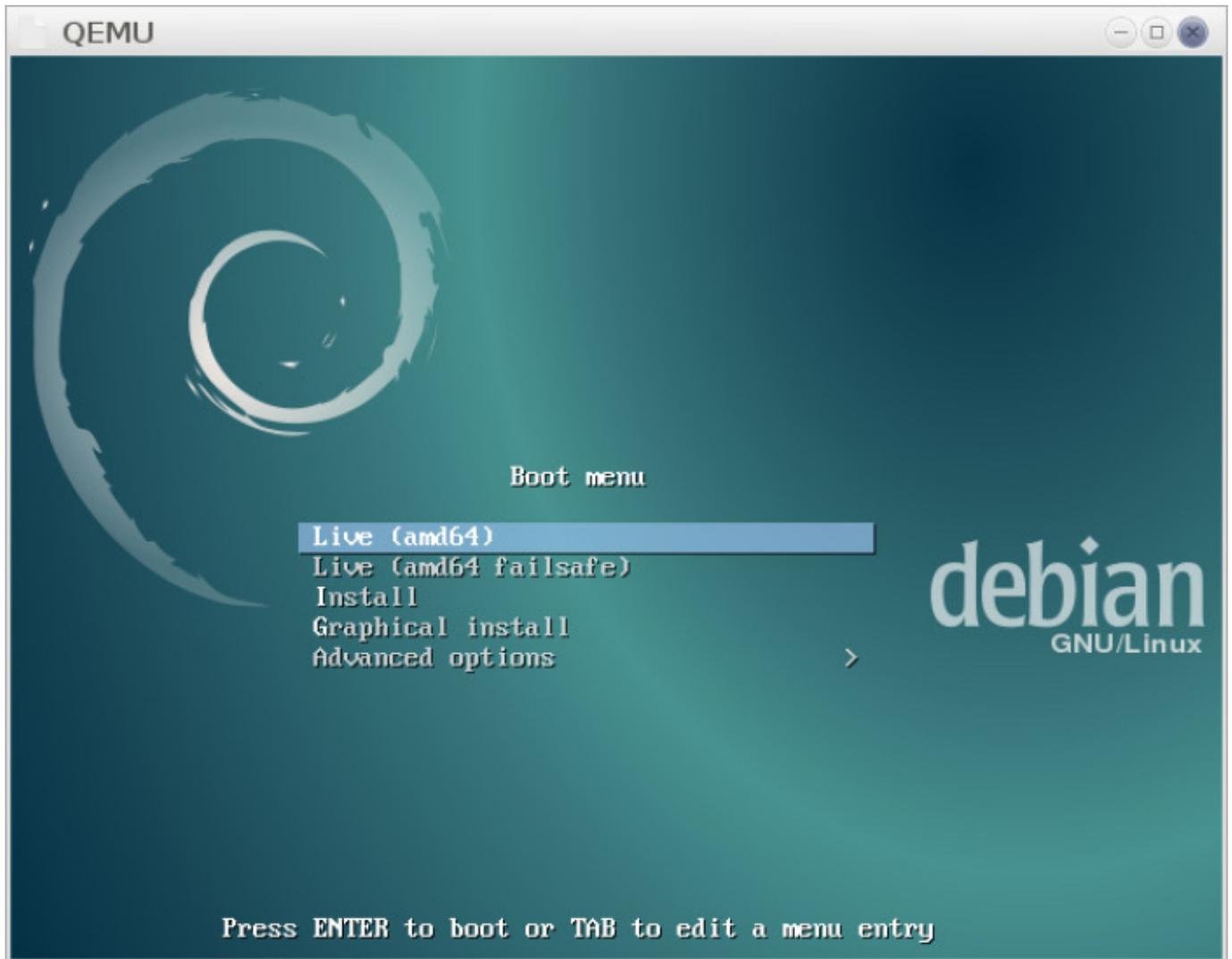
QEMU permite personalizar los componentes de la emulación, tal y como cuando escoge las especificaciones de su computadora portátil.

## Iniciar un sistema operativo

Para ilustrar el ejemplo vamos a ejecutar el live-cd de Debian con el siguiente comando, notese que vamos a cambiar la arquitectura de la emulación a 64bits:

```
qemu-system-x86_64 -cdrom debian-live-8.5.0-amd64-standard.iso
```

Con ello podrá iniciar el sistema operativo Debian GNU/Linux:



## Configuración básica de QEMU

Si esta usando el Live CD podrá notar que la emulación muy lenta y que tal vez requiera de memoria RAM o mas CPUs. En la imagen siguiente se puede notar un error en el kernel por falta de memoria:

```
QEMU
[ 10.243913] Starting init: /bin/sh exists but couldn't execute it (error -28)
[ 10.244657] Kernel panic - not syncing: No working init found. Try passing i
nit= option to kernel. See Linux Documentation/init.txt for guidance.
[ 10.245494] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 3.16.0-4-amd64 #1 Debia
n 3.16.7-ckt25-2
[ 10.245701] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Debia
n-1.8.2-1 04/01/2014
[ 10.246294] 0000000000000000 ffffffff8150e835 ffffffff81705680 ffff8800054bb
f40
[ 10.246893] ffffffff8150b5f8 ffffffff00000008 ffff8800054bbf50 ffff8800054bb
ef0
[ 10.247210] ffff8800054bbef8 0000000000000046 0000000000000817 0000000000000
817
[ 10.247602] Call Trace:
[ 10.248183] [] ? dump_stack+0x5d/0x78
[ 10.248183] [] ? panic+0xc8/0x206
[ 10.248183] [] ? rest_init+0x80/0x80
[ 10.248183] [] ? kernel_init+0xe2/0xf0
[ 10.248183] [] ? ret_from_fork+0x58/0x90
[ 10.248183] [] ? rest_init+0x80/0x80
[ 10.248183] Kernel Offset: 0x0 from 0xffffffff81000000 (relocation range: 0xf
fffffff80000000-0xffffffff9fffffff)
[ 10.248183] ---[ end Kernel panic - not syncing: No working init found. Try
passing init= option to kernel. See Linux Documentation/init.txt for guidance.
```

Modifique el comando para agregarle 2G de memoria RAM a la emulación:

```
qemu-system-x86_64 -cdrom debian-live-8.5.0-amd64-standard.iso -m 2048
```

Ahora sí el sistema iniciará correctamente:

```
QEMU
Debian GNU/Linux 8 debian tty1
debian login: user
Password:
Linux debian 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2 (2016-04-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

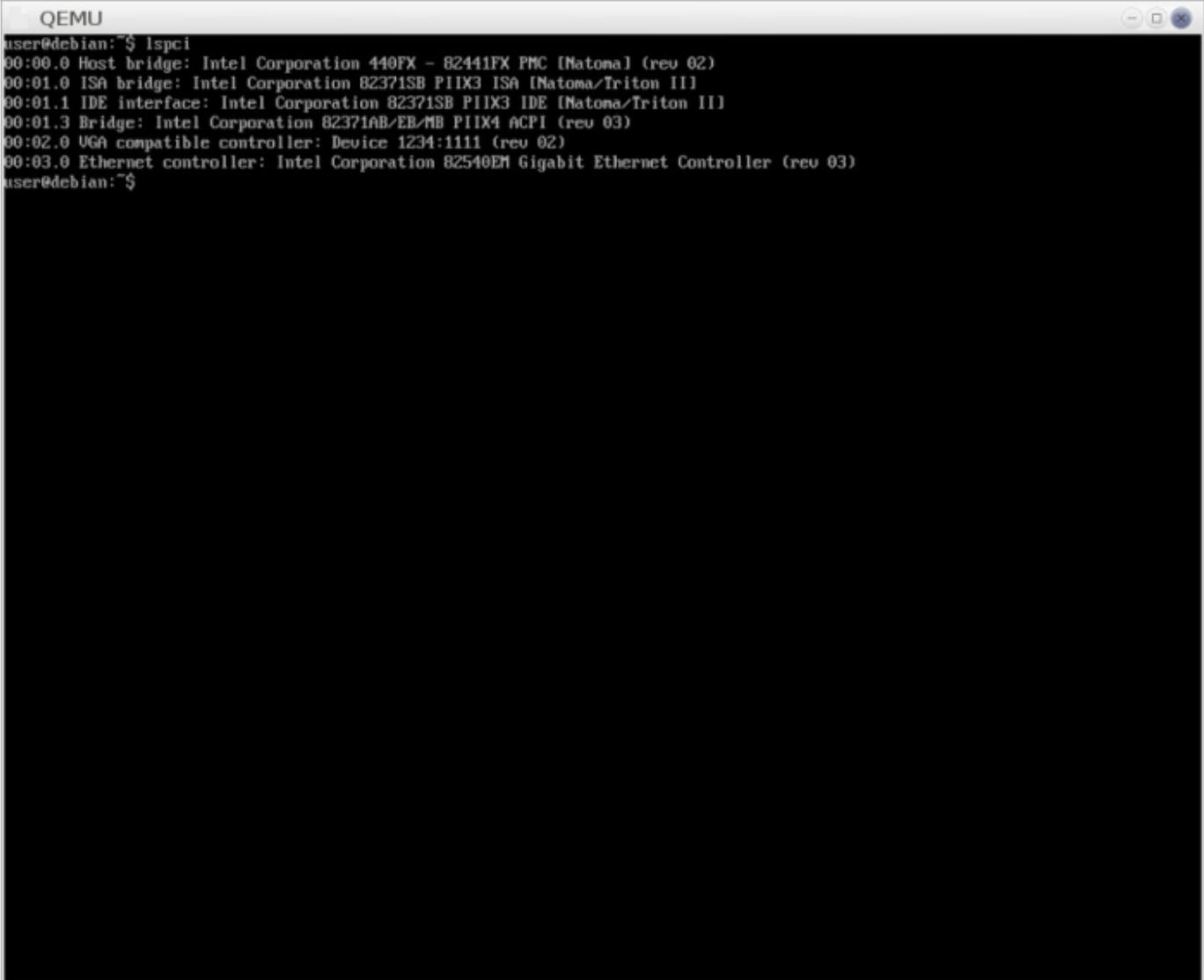
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@debian:~$ _
```

Con el sistema emulado y corriendo vamos a revisar algunas cosas, por ejemplo que CPU piensa el sistema operativo que esta usando:

```
QEMU
user@debian:~$ cat /proc/cpuinfo
processor       : 0
vendor_id     : AuthenticAMD
cpu family    : 6
model         : 6
model name    : QEMU Virtual CPU version 2.5+
stepping     : 3
cpu MHz       : 1497.206
cache size   : 512 KB
physical id   : 0
siblings     : 1
core id      : 0
cpu cores    : 1
apicid       : 0
initial apicid : 0
fpu          : yes
fpu_exception : yes
cpuid level  : 13
wp           : yes
flags        : fpu de pse tsc nsr pae mce cx8 apic sep ntrr pge nca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx lm no
pl pni cx16 hypervisor lahf_lm svm ummcall
bogomips     : 2994.41
TLB size     : 1024 4K pages
clflush size : 64
cache_alignm : 64
address sizes : 40 bits physical, 48 bits virtual
power management:

user@debian:~$ _
```

Tarjeta de video y de red:



```
QEMU
user@debian:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton III]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton III]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Device 1234:1111 (rev 02)
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 03)
user@debian:~$
```

Vamos a configurar el CPU, podemos ver una lista de posibles emulaciones del CPU con el siguiente comando:

```
qemu-system-x86_64 -cpu ?
```

Así que si desea usar la emulación nativa de QEMU ejecute el siguiente comando:

```
qemu-system-x86_64 -cpu qemu64 -m 2048 -cdrom debian-live-8.5.0-amd64-standard.iso
```

Si desea agregar mas cores de CPU a la emulación utilice el parámetro `-smp`, en el siguiente ejemplo se van a asignar cuatro cores:

```
qemu-system-x86_64 -cpu qemu64 -smp 4 -m 2048 -cdrom debian-live-8.5.0-amd64-standard.iso
```

En la siguiente imagen se muestra un extracto de la salida de la comprobación de información del CPU que ilustra los cuatro cores:

```

QEMU
cache size      : 512 KB
physical id    : 2
siblings      : 1
core id       : 0
cpu cores     : 1
apicid        : 2
initial apicid : 2
fpu           : yes
fpu_exception : yes
cpuid level   : 13
mp            : yes
flags         : fpu de pse tsc msr pae mce cx8 apic sep ntrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx 1
pl pni cx16 hypervisor lahf_lm smm ummcall
bogomips      : 3081.47
TLB size      : 1024 4K pages
clflush size  : 64
cache_alignment : 64
address sizes  : 40 bits physical, 48 bits virtual
power management:

processor      : 3
vendor_id     : AuthenticAMD
cpu family    : 6
model         : 6
model name    : QEMU Virtual CPU version 2.5+
stepping     : 3
cpu MHz       : 1497.186
cache size    : 512 KB
physical id   : 3
siblings     : 1
core id      : 0
cpu cores    : 1
apicid       : 3
initial apicid : 3
fpu          : yes
fpu_exception : yes
cpuid level  : 13
mp           : yes
flags        : fpu de pse tsc msr pae mce cx8 apic sep ntrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx 1
pl pni cx16 hypervisor lahf_lm smm ummcall
bogomips     : 3035.47
TLB size     : 1024 4K pages
clflush size : 64
cache_alignment : 64
address sizes : 40 bits physical, 48 bits virtual
power management:

```

Para configurar la tarjeta de video agregue el parámetro -vga, cuidado porque no se despliega una lista de las posibles opciones, no obstante estas son algunas:

- cirrus: es la tarjeta de video mas sencilla y todos los sistemas operativos modernos tienen un driver
- std: características similares a la anterior pero soporta mayores resoluciones de video
- vmware: es una tarjeta de video con mayores capacidades, requiere de instalar los drivers de vmware
- qxl: es una tarjeta de video con mayores capacidades que la std y cirrus utilizando el protocolo SPICE, una tecnología de acceso remoto para máquinas virtuales

Así que para usar la tarjeta de video std utilice el siguiente comando:

```
qemu-system-x86_64 -cpu qemu64 -smp 4 -m 2048 -vga std -cdrom debian-live-8.5.0-amd64-standard.iso
```

Los parámetros de CPU y video que especificamos son los que vienen por defecto, no es necesario indicarlos, se hace a manera de exponerlo.

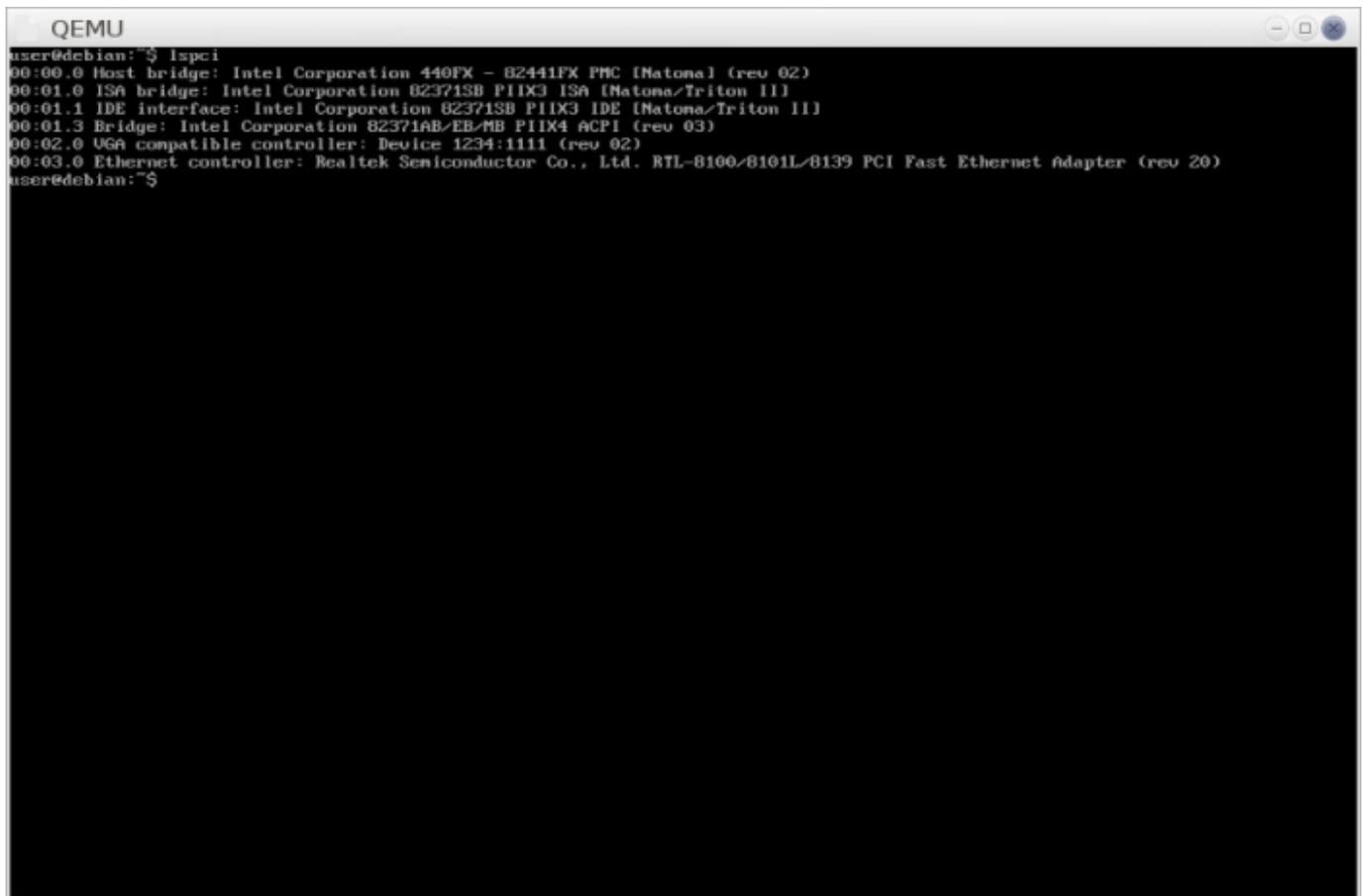
Para ver las posibles configuraciones de las tarjetas de red podemos usar el comando:

```
qemu-system-x86_64 -net nic,model=help
```

En el siguiente ejemplo se va utilizar el modelo rtl8139 de Realtek, el comando a utilizar es el siguiente:

```
qemu-system-x86_64 -net nic,model=rtl8139 -cpu qemu64 -smp 4 -m 2048 -vga
std -cdrom debian-live-8.5.0-amd64-standard.iso
```

Revisamos el cambio del modelo de la tarjeta de red:



```
QEMU
user@debian:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 UGA compatible controller: Device 1234:1111 (rev 02)
00:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8100/8101L/8139 PCI Fast Ethernet Adapter (rev 20)
user@debian:~$
```

## TODO

El parámetro -display

## Referencias

- <https://wiki.gentoo.org/wiki/QEMU/Options>
- <http://wiki.qemu.org/Manual>
- <http://qemu-buch.de/de/index.php?title=Hauptseite>

From:  
<https://www.estebanmonge.site/> - **Esteban Monge**

Permanent link:  
<https://www.estebanmonge.site/doku.php?id=qemu>

Last update: **2016/06/15 12:57**

