

Cuestiones 4

Administración de sistemas II

3er Curso, 2º Cuatrimestre

Grado de Ingeniería Informática

Pablo Doñate Navarro

Adnana Dragut

Año

2021/2022



TABLA DE CONTENIDOS

[**Cuestiones resueltas**](#_heading=h.gjdgxs) **3**

# **Cuestiones resueltas**

**Cuestión 1.** Enunciar y detallar el emulador de máquinas de código abierto QEMU.

Es un emulador de procesadores que se encarga de la traducción del código binario de la arquitectura fuente en código comprensible por la arquitectura huésped. El emulador QEMU soporta la virtualización tanto dentro de un sistema operativo Linux como Windows. QEMU puede ejecutarse en cualquier tipo de arquitecturas como puede ser PowerPC, MIPS o SPARC.

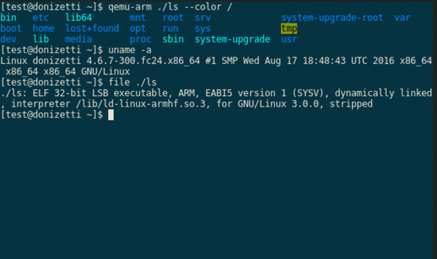
El principal fin de este emulador es el de permitir ejecutar sistemas operativos en arquitecturas distintas a aquellas para las cuales fueron inicialmente desarrolladas. Además, para que pueda funcionar correctamente necesita poseer módulos de kernel KVM o KQEMU.

El emulador no cuenta con interfaz gráfica, sin embargo, se han implementado programas específicos para QEMU capaces de permitir al usuario interaccionar mediante una GUI. En el caso de Windows este programa se denomina “QEMU manager” y en el caso de Linux este programa se denomina “qemu-launcher”.

Un ejemplo de aplicación de QEMU es la emulación de una computadora japonesa denominada NEC PC-9801.

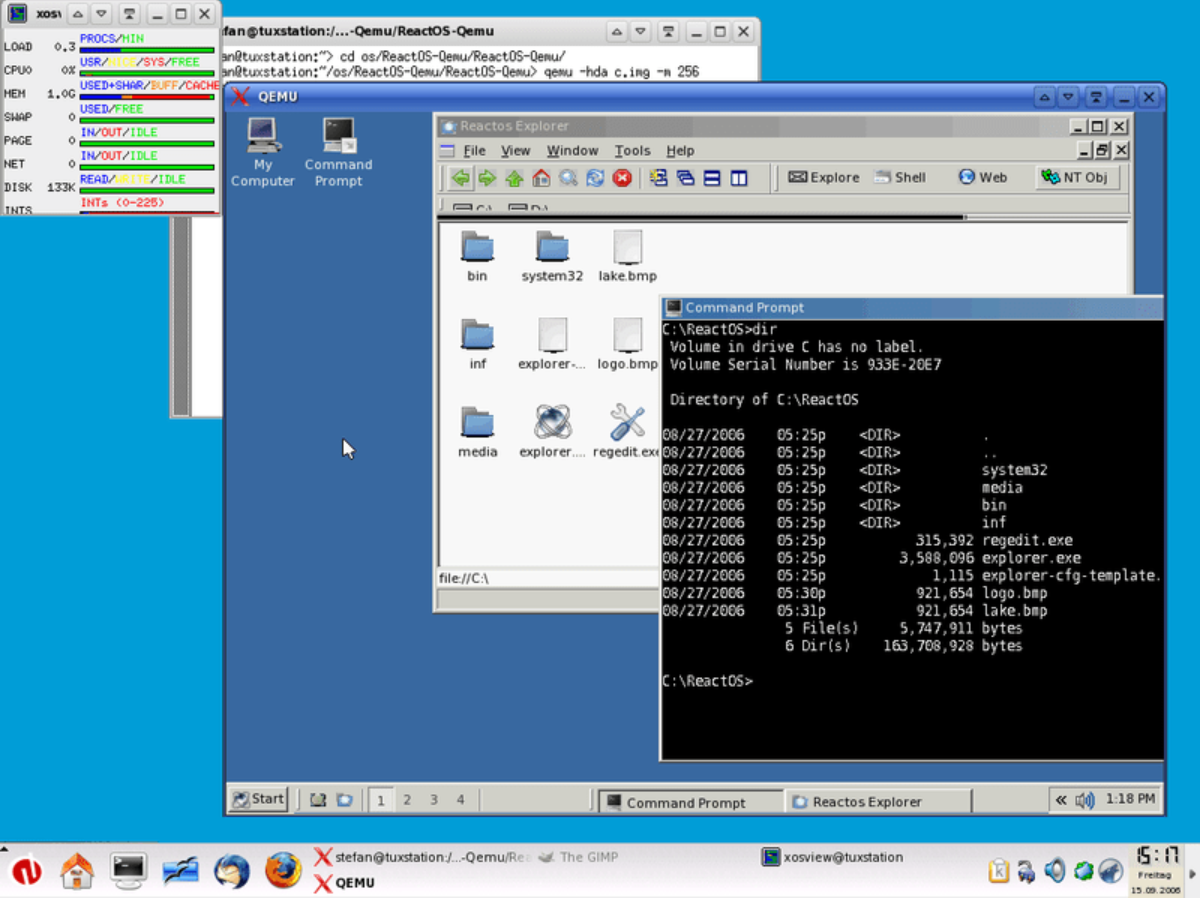
QEMU cuenta con dos modos de funcionamiento:

* El modo usuario → Modo de funcionamiento capaz de ejecutar programas diseñados para un tipo de CPU en otra CPU diferente.



*Figura 1. Ejemplo de emulación en modo usuario*

* El modo de emulación completo de sistema de ordenador → Modo de funcionamiento que permite emular todas las características de un sistema informático, así como el procesador y varios periféricos.



*Figura 2. Ejemplo de emulación en modo completo*

Las principales características de QEMU son:

* Soporte de emulación de diversas arquitecturas como IA-32, AMD64, MIPS o Sun SPARC.
* Permite que ciertas aplicaciones tengan una velocidad semejante a la que se obtendría si la aplicación se hubiera ejecutado en un sistema real.
* Posee formato en disco Copy On Write.
* Capacidad de salvar y restaurar el estado de la máquina.
* Permite la emulación de tarjetas de red.
* Soporta el SMP (multiprocesamiento simétrico).
* Permite un control remoto de la máquina emulada mediante un servidor VNC.

**Cuestión 2.** Enunciar y describir el paquete Xen-tools.

El paquete Xen-tools hace referencia a un conjunto de scripts, escritos en lenguaje Perl, que permiten gestionar los servidores virtuales Xen basados en la distribución de linux llamada Debian. Así mismo, Xen-tools es utilizado en las pruebas de sistema automatizadas de “XenProject.org”, un proyecto encargado de mejorar la virtualización en diferentes ámbitos comerciales y aplicaciones de código abierto.

Xen-tools permite crear, modificar y eliminar dominios huésped Xen. Los tipos de invitados que pueden ser gestionados con este paquete son los PV, que hacen referencia a la paravirtualización, y los invitados HVM, que hacen referencia a una virtualización completa. Xen-tools permite acceder a dichos invitados por medio de conexiones SSH.

A diferencia de los programas ofrecidos por otras máquinas virtuales, Xen-tools está basado en chroots, que los desarrolladores utilizan para poder crear paquetes con programas para diferentes versiones de un sistema operativo. Los usuarios también utilizan estos chroots para poder acceder a las distintas versiones de paquetes desarrollados.

La principal desventaja de este paquete es que puede gestionar un número limitado de invitados.

**Cuestión 3.** Enunciar y describir el proyecto Libvirt.

Libvirt hace referencia a un conjunto de software que permite gestionar máquinas virtuales, almacenamiento virtual, así como interfaces de red virtuales. Libvirt está formado por un API escrita en lenguaje C, un demonio denominado “libvirtd”, y la herramienta virsh utilizada para poder administrar huéspedes virtuales a través de línea de comandos.

El objetivo principal que persigue este proyecto es el de permitir gestionar de forma conjunta diferentes hipervisores y máquinas virtuales como QEMU, LXC, Xen, OpenVZ, el hipervisor de VirtualBox, Vmware, Hyper-V o PowerVM.

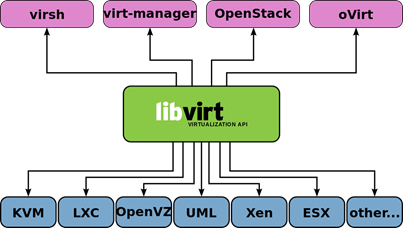
Aunque la biblioteca libvirt está escrita en lenguaje C, también permite el acceso a otros lenguajes de programación como Python, Perl, Ruby, Java o JavaScript

Las características fundamentales de Libvirt son las siguientes:

* Permite gestionar máquinas virtuales en base a órdenes de arranque, parada, pausa, restauración y migración.
* Soporta máquinas remotas a las que se puede acceder mediante conexiones SSH.
* Permite gestionar distintos tipos de almacenamiento por medio del demonio de libvirt.

* Permite gestionar tanto interfaces de red físicas como interfaces de red lógicas.
* Soporta redes virtuales de tipo NAT.

En el esquema inferior se muestra la relación de libvirt con otros gestores, máquinas virtuales e hipervisores existentes.



**Cuestión 4.** Enuncia y describe la suite virt-manager

El programa virt-manager, también conocido como gestor de máquina virtual Red Hat, es una aplicación de escritorio que tiene como fin principal el de gestionar máquinas virtuales a través de la librería libvirt. Fue desarrollado por Red Hat y está escrito en el lenguaje de programación Python. Está dirigido a las máquinas virtuales KVM, al hipervisor de Xen y al contenedor de linux LXC.

Esta aplicación posee wizards que permiten controlar el ciclo de vida de las distintas máquinas virtuales, la creación de dominios, la configuración de recursos, así como la gestión del hardware virtual.

Las principales funcionalidades de virt-manager son las siguientes:

* Permite crear, editar, iniciar, pausar, reanudar y parar máquinas virtuales.
* Permite ver y controlar las consolas de cada máquina virtual.
* Permite monitorizar todas las máquinas virtuales que están en ejecución.

* Permite ver el porcentaje de uso y el rendimiento de cada máquina virtual.

* Permite administrar máquinas virtuales que se ejecutan de forma remota.

* Permite ejecutar contenedores LXC.

* Permite gestionar diferentes tipos de almacenamiento.

* Permite gestionar distintas interfaces de red.

Algunos de los comandos que se instalan con el paquete virt-manager son los siguientes:

* virt-install → permite la creación de nuevas máquinas virtuales a través de libvirt.

* virt-clone → permite la duplicación de máquinas virtuales existentes a través de libvirt.
* virt-xml → permite modificar máquinas virtuales existentes a través de libvirt.
* virt-viewer → es un interfaz de usuario ligera, a través de la cual se puede interaccionar con el sistema operativo virtualizado.
* virt-bootstrap → es una línea de comandos que permite configurar ficheros root para los contenedores basados en libvirt.

**Cuestión 5.** Enuncia y describe la suite Bonnie++

Bonnie++ es una herramienta utilizada para analizar el rendimiento de discos duros, así como de sistemas de archivos. Permite la creación de tests de lectura, escritura y borrado de archivos.

El análisis puede realizarse desde la terminal, y por tanto los resultados del mismo pueden ser observados desde ella o ser direccionados a un fichero, que puede ser exportado a formato csv.

La órden bonnie++ puede recibir diferentes parámetros, entre los que destacan los siguientes:

* -n: para indicar el número de pruebas de creación de archivos a realizar.
* -u: para indicar con qué permisos se va a ejecutar.
* -r: para indicar el tamaño de la RAM en MB.
* -f: para indicar si se tienen que realizar las pruebas de carácter de E/S.
* -b: para indicar si se va a forzar la escritura en disco en lugar de caché.
* -d: para indicar el directorio que contendrá los archivos creados.

**Cuestión 6**. Enuncia y describe la funcionalidad de QEMU en KVM.

KVM (Kernel Virtual Machine) es una solución de virtualización para linux. Con él, se puede convertir Linux a un hipervisor para que un host ejecute entornos virtuales.

QEMU es un emulador de procesadores basado en traducción dinámica de binarios, es decir, realiza la conversión del código binario de la arquitectura host en código entendible por la arquitectura huésped.

KVM es el módulo del kernel de Linux que permite la asignación de CPU físico para CPU virtual. Esta asignación proporciona la aceleración de hardware para la máquina virtual y aumenta su rendimiento. De hecho QEMU utiliza esta aceleración cuando el tipo de virtualización es elegido como KVM.

Los desarrolladores de KVM crearon un nuevo modelo de CPU en QEMU. En dicho modelo, las llamadas al sistema pasan por el módulo KVM para que la ejecución se ejecute de forma nativa en la CPU, mientras que QEMU se utiliza para proporcionar el resto de funcionalidad. Al trabajar juntos, KVM accede directamente al CPU físico y a la memoria, a su vez QEMU emula los recursos de hardware, como el disco duro, video, USB, etc.

**Cuestión 7.** Enuncia y describe los dos subtipos de hipervisores de tipo 1 (monolíticos, de microkernel).

Antes de nada, vamos a comentar lo que son los hipervisores de tipo 1. Después veremos los dos subtipos que tenemos:

Los hipervisores de tipo 1 son aquellos que el hipervisor se ejecuta directamente sobre el hardware físico; el hipervisor se carga antes que ninguno de los sistemas operativos invitados, y todos los accesos directos a hardware son controlados por él.

Los hipervisores monolíticos son aquellos que emulan el hardware para sus máquinas virtuales. Esto es posible debido a que hay una gran cantidad detrás que se interpone entre los recursos físicos reales y las operaciones de acceso a ellos efectuadas por las máquinas virtuales.

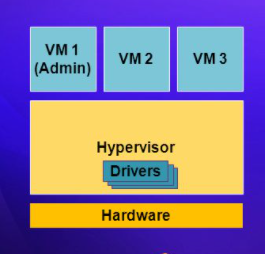
El proceso que sigue una llamada a hardware en un sistema virtualizado usando un hipervisor de tipo monolítico es:

1) El hardware emulado debe interceptar la llamada.

2) El VMM redirige estas llamadas hacia los drivers de dispositivo que operan dentro del hipervisor, lo cual requiere de numerosos cambios de contexto en el código de la llamada.

3) Los drivers del hipervisor dirigen o enrutan la llamada hacia el dispositivo físico.

Debido a este funcionamiento, se obliga a desarrollar drivers específicos para el hipervisor de cada componente hardware.



Los hipervisores de microkernel son aquellos en los cuales los hipervisores se reducen a una capa de software muy sencilla, cuya única funcionalidad es la de particionar el sistema físico entre los diversos sistemas virtualizados.

Con esta manera de funcionar los hipervisores de microkernel no requieren de drivers específicos para acceder al hardware.

Gracias a este tipo de hipervisores se aumenta el rendimiento al reducir el código intermedio y el número de cambios de contexto necesarios, también aumentan la estabilidad de los sistemas, al haber menos componentes, y sobre todo la seguridad, al reducir la superficie de ataque del hipervisor.

