

Virtualización y emulación en Linux

# VIRTUDES DE LO VIRTUAL

Encontrarás una solución de virtualización en cada medio Linux, desde el escritorio al servidor de la empresa. En la sección de portada de este mes investigamos algunas herramientas de virtualización para usuarios de Linux muy prometedoras. **POR TIM SCHÜRMANN.**

**E**n los viejos tiempos, un sistema operativo era, en cierto modo, algo atado de forma monógama al hardware. Excepto algunos expertos y visionarios, nadie consideraba siquiera la posibilidad de varios sistemas compartiendo máquina de forma simultánea. Y nadie perdía el sueño preocupándose por la opción de un sistema operativo funcionando dentro de otro sistema diferente. En el mundo virtual de hoy, sin embargo, tus aplicaciones quizá no sepan dónde termina el hardware y dónde empieza el software.

La virtualización ofrece numerosos beneficios para el usuario de Linux: estabilidad, manejabilidad, seguridad e incluso nostalgia. Este mes miraremos de cerca algunas opciones de virtualización para Linux.

Comenzaremos comparando las alternativas de virtualización para el escritorio Linux; después las examinaremos en servidores IBM System p. Aprenderás incluso sobre el famoso emulador Bochs, y acabaremos con un vistazo a la herramienta IEs4Linux, basada en Wine, que te permite utilizar Internet Explorer bajo Linux.

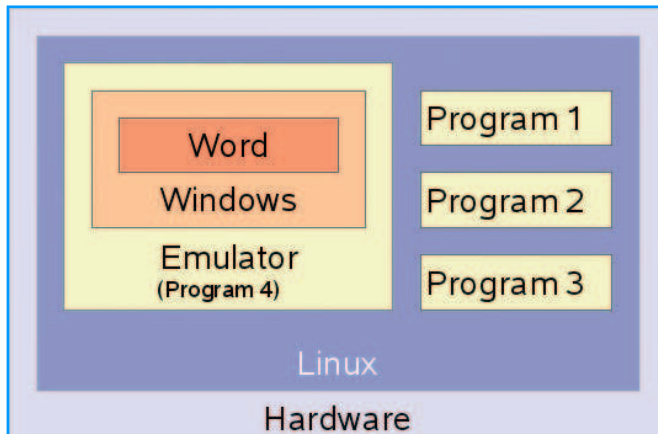
## Competición y Emulación

Hace más de 40 años, mucho antes de que se inventara Linux, IBM tenía un problema. El nuevo System /360 [1] funcionaba de un modo totalmente distinto al del anticuado



### EN PORTADA

Herramientas de Virtualización . . . . .	21
Virtualización en System p . . . . .	28
El Emulador Bochs . . . . .	31
IEs4Linux . . . . .	35



**Figura 1: Un emulador crea un PC dentro de un PC. En este ejemplo, Windows y Word están funcionando dentro de la emulación. Ninguna de las dos aplicaciones puede acceder al sistema Linux subyacente.**

7070. Para facilitar la migración de los clientes potenciales, IBM quería que las aplicaciones antiguas funcionaran en el nuevo sistema.

Tras varias pruebas, IBM se decantó finalmente por una idea ofrecida por el ingeniero Larry Moss, el cual sugirió una combinación de software y una extensión de hardware especial. La solución combinada monitorizaría cada paso de la aplicación antigua y convertiría sus comandos en otros que el nuevo System /360 pudiera entender. De este modo, el nuevo ordenador se comportaría exactamente igual que su predecesor. Este sencillo juego de manos permitió a los compradores del nuevo modelo la posibilidad de continuar usando sus aplicaciones 7070. Dado que este software emulaba el comportamiento de otro

más genéricamente, una “máquina virtual” (véase la Figura 1). QEMU [2] y Bochs [3] son sólo dos ejemplos de emuladores populares y gratuitos, que simplemente reenvían la salida de pantalla a una ventana de aplicación propia. El sistema operativo que funciona en la máquina virtual cree que está usando una pantalla estándar.

Los emuladores ofrecen un buen número de beneficios. Para empezar, es bastante simple clonar un sistema funcionando en el emulador. Dado que éste copia un PC completo a nivel de software, es bastante sencillo congelar el estado de un momento dado y crear una captura de pantalla, o hacer que el PC virtual funcione en otra computadora totalmente distinta.

ordenador, Larry Moss bautizó su invento como “emulador”.

Los emuladores modernos copian el comportamiento de una máquina completa a nivel de software, y lo hacen tan a la perfección que puedes lanzar en un emulador un sistema operativo invitado junto a las aplicaciones diseñadas para él.

Esto ofrece al usuario un PC virtual, o

Puesto que el hardware simulado no cambia, puedes incluso cruzar las fronteras del hardware. Esto permite que el usuario pruebe nuevos programas sin ningún riesgo: sólo salvas el estado actual del emulador, instalas la aplicación que quieres probar y entonces restauras el estado original si no estás satisfecho con el resultado de la prueba.

Además de estas soluciones exclusivamente software, también existen emuladores hardware. Y, como es el caso del histórico IBM System /360, además hay soluciones híbridas. Un ejemplo reciente de una solución híbrida es la PlayStation 3 de Sony. Su primera versión incluía un componente hardware que permitía que los juegos de PlayStation 2 funcionaran en la consola. La actual versión europea de PS3 emplea un emulador exclusivamente software para lo mismo.

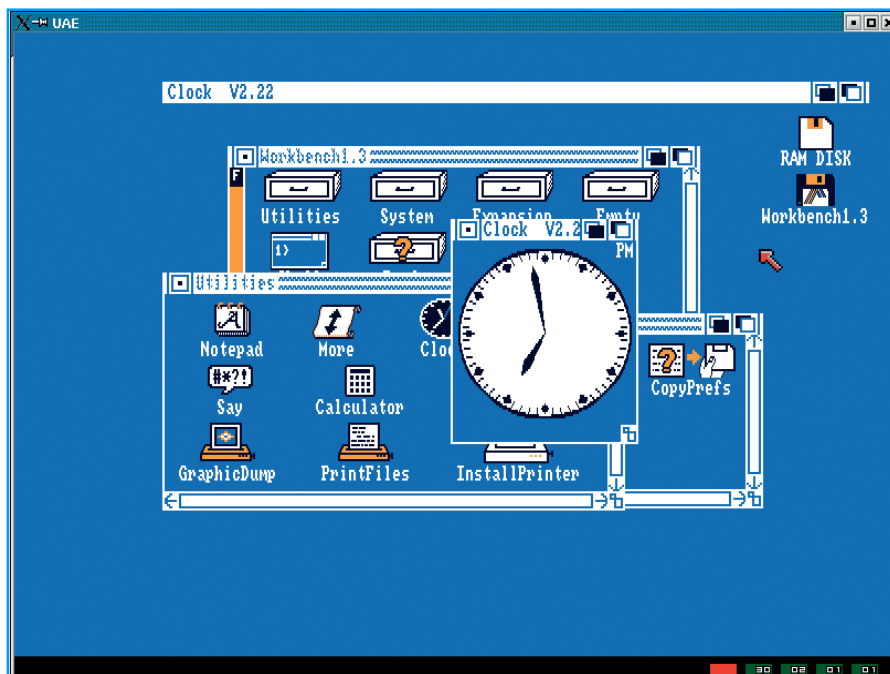
Este ejemplo muestra otro uso principal de los emuladores: la capacidad de insuflar nueva vida en hardware y software ya

### Máquinas Virtuales Desconocidas

Las soluciones de virtualización brotan en los lugares más inesperados. Por ejemplo, la alternativa BIOS de Open Firmware es a la vez una máquina virtual [16]. El hardware instalado en el ordenador puede almacenar sus propias extensiones de la configuración básica en él, sin importar sobre qué tipo de plataforma hardware funciona.

Algunas CPUs u ordenadores soportan un modo de compatibilidad. Todas las CPUs x86 actuales pueden utilizar programas de sus antecesores. El fabricante de CPU Transmeta avanza un paso más en esta dirección: para mantener la compatibilidad con Intel y AMD, Transmeta traduce *todas* las instrucciones a su propio formato antes de ejecutar.

El Commodore 128 de los 80 es otro ejemplo de virtualización. El ordenador casero tenía un modo de operación nativo, pero también podía emplear aplicaciones diseñadas para su hermano pequeño, el C-64, y para CP/M, el sistema operativo de PC mejor acabado de la época.



**Figura 2: El emulador UAE revive el viejo Commodore Amiga. Aquí puedes ver el GUI de Amiga funcionando en una ventana de Linux.**

### Glosario

**Emulador:** Programa que copia el comportamiento de otra aplicación, derivado de la palabra latina “aemulare” (competir, simular).

difunto. Simplemente, imagínate utilizando los buenos y viejos programas de cuentas de los gloriosos días de DOS o el legendario procesador WordPerfect de la misma era.

Los emuladores de este tipo tienden a estar enfocados hacia las videoconsolas y ordenadores caseros del último siglo. Los fans han creado emuladores que copian estos tesoros heredados con precisión de bits. Esto da una segunda oportunidad, en PCs Linux, a máquinas clásicas como el Commodore Amiga [4], Atari ST [5] o el C-64 [6].

Dado que estos programas son lo que son, pueden causar algunas complicaciones. Un emulador necesita copiar los componentes hardware de un ordenador con tanta precisión como sea posible, así que tendrás que asegurarte de que el sistema dispone de suficiente potencia computacional.

Los viejos sistemas, construidos para hardware menos avanzado, suelen ser menos exigentes. Como regla general, cuanto más viejo es el sistema operativo, más rápido correrá en un emulador.

Contrastando con esto, una versión reciente de Ubuntu en QEMU funciona como si estuvieras empleando Ubuntu en una máquina que ha visto días mejores.

Cuanto más lo piensas, menos sentido tiene emular una CPU Intel completa si ya tienes una en tu ordenador. En su lugar, puedes emplearla directamente, y tendrás que simular sólo el resto de tu sistema objetivo.

Puedes aplicar el mismo principio a cualquier otro componente simplemente emulando el hardware que necesitas a nivel software. En este caso, ya no nos referimos al método como emulación, sino como virtualización.

Si le quitas la simulación de CPU a un emulador, se convierte en una virtualización de hardware o de sistema. Puedes instalar la extensión KQEMU de QEMU para emplear este modo. El producto comercial VMware [7] también funciona sin emulación de CPU. El punto negativo de este método es la restricción a un tipo específico de CPU. Por ejemplo, ninguna de estas opciones funcionará en un PowerPC recién sacado de la caja.

## Paravirtualización

Un emulador funciona como un programa normal en Linux o Windows, pero teóricamente podrías eliminar el sistema operativo anfitrión. Para que esto pueda ocurrir, nece-



Figura 3: ScummVM trae de vuelta a la vida a viejas aventuras dándoles lo que necesitan: un intérprete de código.

sitas software especial que funcione directamente sobre el hardware.

Los programas de este tipo se conocen como hipervisores o monitores de máquinas virtuales. El hipervisor gestiona los sistemas operativos huéspedes independientemente del sistema operativo, lo que asegura que haya operaciones paralelas sin problemas.

En este tipo de medios, los sistemas huéspedes no trabajan directamente con el hardware, sino que envían sus peticiones al hipervisor. Por ejemplo, si uno de los sistemas Linux huésped funciona en paralelo necesita acceso al disco, envía una petición de acceso al hipervisor. Éste maneja, entonces, el acceso físico, y devuelve los resultados al sistema.

Para permitir que todos los invitados “hablen” con el hipervisor, éste les ofrece un interfaz estandarizado para el hardware físico, que podrán usar otros programas y sistemas. Esta técnica, conocida como paravirtualización, tiene la ventaja de una ejecución sorprendentemente rápida comparada con otras soluciones.

Los vendedores de soluciones de paravirtualización ofrecen un índice de fallos de sólo un 0,5 ó 3 por ciento comparado con el hardware físico. El producto gratuito Xen [8] y el comercial ESX Server de VMware [7] son los ejemplos más populares de esta tecnología.

Los requisitos del sistema huésped para soportar el hipervisor son un obstáculo para la paravirtualización, ya que implican modificar el sistema operativo. Con un sis-

tema tan blindado como Microsoft Windows, esta tarea es obviamente difícil. Otra complicación es que el hipervisor tiene que manejar personalmente numerosas tareas del sistema operativo. Por ejemplo, tiene que saber qué tipo de adaptador de gráficos tiene el sistema y cómo dirigirse a él.

Para evitar naufragar en un mar de modificaciones de drivers, los desarrolladores de hipervisores suelen optar por uno de los métodos siguientes:

- El hipervisor del proyecto Xen simplemente elige uno de los sistemas operativos paralelos como su huésped favorito. Una vez que se ha establecido esta relación, emplea los drivers de ese sistema. En otras palabras, si algún otro SO funcionando en la máquina accede a un interfaz USB, el hipervisor le pasa la petición a su favorito.
- El segundo método transforma un kernel existente de Linux para producir un hipervisor (Linux tiene, más o menos, todo lo que éste necesita). El proyecto KVM [9], por ejemplo, emplea la técnica de la virtualización; provee de un módulo kernel que convierte al del Linux en uso en un hipervisor, que a su vez emplea un emulador modificado a partir de QEMU para lanzar otros sistemas operativos. A este método de virtualización se le llama “basado en kernel”.

## Virtualizando Sistemas Operativos

Si sólo necesitas clonar una instancia de Linux que está funcionando en tu máquina,

tu mejor opción probablemente sea la virtualización de sistemas operativos. Al contrario que en otras soluciones de virtualización, ésta consiste en que el ordenador sólo debe hacer funcionar un sistema, que se clona según unas plantillas, las cuales especifican la configuración para el nuevo clon. Por ejemplo, los administradores pueden restringir el espacio al disco duro.

En la clonación, la virtualización normalmente sólo copia el medio del sistema. En el fondo sigue habiendo nada más que un kernel de Linux, que todos los programas activos emplean, sin importar en qué clon están funcionando. Lo que hace este método es encerrar una aplicación en su propio medio de sistema, lo que lo convierte en algo rápido y seguro. Sin embargo, no se soportan programas de otros sistemas operativos.

Linux VServer [10] y OpenVZ [11] un derivado en código abierto del comercial Virtuozzo, son ejemplos de virtualizadores de sistemas operativos gratuitos. Ambas variantes emplean un kernel de Linux modificado.

### Licencias

Para utilizar un sistema operativo comercial en una máquina virtual necesitas una licencia válida. Por ejemplo, para emplear un sistema Microsoft en un entorno virtual, tienes que comprar tu propia copia de Windows. Comprueba los detalles de la licencia del fabricante de tu software para andar sobre seguro.

Otra cuestión (y una que afecta particularmente a los ordenadores o sistemas operativos antiguos) es el firmware patentado, que podría estar protegido por el copyright o por leyes de patentes. Por ejemplo, necesitas una BIOS compatible para una emulación de PC completa. Las máquinas Apple antiguas incluso almacenan parte del sistema operativo en chips, en un método muy popular en esos días de computadoras caseras. En el caso del PC, afortunadamente, hay varias BIOS gratuitas para elegir [17], y la mayoría de productos de virtualización incluye alguna. Para el resto de ordenadores, no hay más alternativa que obtener el firmware mediante un programa especial.

## Virtualización de Aplicaciones

En algunos casos es innecesario clonar un sistema completo para mantener separados los programas. La virtualización de aplicaciones ofrece al usuario un paquete libre de problemas para cualquier aplicación, que contiene los recursos que ésta necesita para funcionar.

Entre esos recursos pueden incluirse archivos de configuración, librerías o programas auxiliares. El paquete que se crea así se llama entorno (virtual) de ejecución.

No hace falta instalar los programas empaquetados de este modo. Las aplicaciones desarrolladas en lenguaje Java son ejemplos de esta categoría. Para ejecutar un programa Java necesitas el Java Runtime Environment [12]. ScummVM [13], responsable de resucitar unas cuantas aventuras gráficas viejas, también confía en este principio (Figura 3).

## Virtualización Parcial

La virtualización parcial es un caso especial en este mundo. Esta solución implica pretender que un componente de hardware existe varias veces. Si esto se efectúa con la memoria RAM, el proceso es conocido como virtualización parcial. En este caso, el programa que funciona en el sistema se cree que tiene acceso exclusivo a la memoria. Los demás procesos quedan ocultos a la aplicación.

## Virtualización de Servidores

Los proveedores de servicios de Internet están particularmente interesados en las soluciones de virtualización. En primer lugar, estas soluciones soportan hospedaje compartido, es decir, la posibilidad de que múltiples clientes compartan un servidor de Internet físico. Estos clientes asumen (y no están totalmente equivocados) que están trabajando en sus propios servidores. Los operadores de un sitio web pueden emplear esta tecnología para optimizar su respuesta a peticiones vía Internet, distribuyendo la carga entre sus servidores, a la vez que reducen costes operativos comparando hardware.

Las soluciones de virtualización empleadas para los servidores confían en los modelos de los que hablábamos antes, pero dependen también de opciones a nivel de servidor, como la distribución de carga o interfaces administrativos especiales.

Como ejemplos de productos para virtualización de servidores tenemos VServer o OpenVZ. Ambos sistemas emplean un kernel de Linux modificado, que lanza y gestiona los medios virtuales individuales.

## Conclusiones

Aparte del gran volumen de palabrería que difunden los técnicos y consultores de marketing, la virtualización ofrece opciones interesantes para un usuario Linux. Puedes hacer funcionar distintos sistemas operativos al mismo tiempo, aislar aplicaciones críticas y probar nuevas configuraciones sin ningún riesgo para los sistemas existentes. Esperamos que hayas disfrutado del reportaje del mes sobre virtualización. ■

### RECURSOS

- [1] IBM System/360: <http://es.wikipedia.org/wiki/System/360>
- [2] Qemu y KQemu: <http://fabrice.bellard.free.fr/qemu/>
- [3] Emulador PC Bochs: <http://bochs.sourceforge.net>
- [4] Emulador UAE Amiga: <http://uae.coresystems.de>
- [5] Emulador StonX Atari ST: <http://stonx.sourceforge.net>
- [6] Emulador VICE Commodore 64: <http://www.viceteam.org>
- [7] VMware: <http://www.vmware.com>
- [8] Xen: <http://www.cl.cam.ac.uk/research/srg/netos/xen/>
- [9] KVM: <http://kvm.qumranet.com/kvmwiki>
- [10] VServer: [http://linux-vserver.org/Welcome\\_to\\_Linux-VServer.org](http://linux-vserver.org/Welcome_to_Linux-VServer.org)
- [11] OpenVZ: <http://openvz.org>
- [12] Lenguaje de programación Java: <http://www.java.com>
- [13] ScummVM: <http://www.scummvm.org>
- [14] Intel Virtualization Technology: <http://www.intel.com/technology/platform-technology/virtualization/index.htm>
- [15] Virtualización AMD: [http://www.amd.com/us-en/Processors/ProductInformation/0,,30\\_118\\_8796\\_14287,00.html](http://www.amd.com/us-en/Processors/ProductInformation/0,,30_118_8796_14287,00.html)
- [16] Open Firmware: <http://www.openbios.info>
- [17] BIOS gratuita para PC: [http://www.linuxbios.org/Welcome\\_to\\_LinuxBIOS](http://www.linuxbios.org/Welcome_to_LinuxBIOS)