

# Capítulo 2

## “Herramientas de Virtualización”

### 2.1 Introducción

Las herramientas para la virtualización permiten crear máquinas virtuales. Una máquina virtual es una aplicación que reproduce el comportamiento de una PC, de un sistema operativo bajo otro de distintas características que emula el *hardware* de una computadora. La máquina virtual puede tener o no la capacidad de ejecutar un sistema operativo diferente al residente en la máquina anfitriona<sup>1</sup> y dentro de ella se pueden ejecutar otras aplicaciones o interpretar lenguajes de programación.

Un emulador es un programa que intenta realizar las funciones de la máquina a la que emula. La emulación es un intento de replicar el diseño interno de un determinado dispositivo que en el contexto de esta tesis es un PC.

De acuerdo al tipo de herramienta que se utilice se puede virtualizar un sistema de *hardware* completo dentro de la máquina anfitriona o solamente un determinado sistema operativo.

### 2.2 Herramientas de Virtualización

En las subsecciones posteriores se realizará una breve reseña de las opciones más utilizadas en cuanto a *software* de virtualización se refiere [2].

#### 2.2.1 User Mode Linux (UML)

Permite iniciar una máquina Linux como un proceso de usuario ejecutándose dentro de una máquina anfitriona. Desde el punto de vista de la máquina anfitriona, UML es un proceso normal de usuario. Un proceso que se ejecuta sobre UML, ve un *kernel* (distinto al del sistema anfitrión) que le proporciona memoria virtual y acceso a los dispositivos físicos (unidades de disco, CD R/W, etc) y/o virtuales (por ejemplo: Interfaz virtual de red) [1][5].

Desde el punto de vista de un usuario bajo linux, UML es similar a otras herramientas de virtualización, entre ellas VMware, sin embargo, existe una diferencia conceptual respecto a las filosofías de diseño entre UML y VMware. Mientras VMware emula el *hardware* de una PC y puede crear instancias virtuales bajo diferentes sistemas operativos (Linux,

---

<sup>1</sup> En el contexto de esta tesis, se denomina anfitrión al sistema operativo que constituye el soporte para todos los procesos incluyendo las máquinas virtuales. Cualquier otro sistema operativo que se ejecute como un proceso de usuario recibe el nombre de huésped o invitado.

Windows, etc.), UML "virtualiza" las llamadas al sistema y sólo puede ser ejecutado bajo Linux. El *kernel* de Linux que se ejecuta y soporta UML es consciente de lo que existe en las capas inferiores, aunque los procesos que ejecuta dicho *kernel* no son conscientes de esta diferencia. La Figura 2. 1 describe un sistema Linux típico mientras que en la Figura 2. 2 se ilustra un sistema operativo Linux que incluye *User Mode Linux* junto a procesos de usuario paralelos y otros en una capa superior.

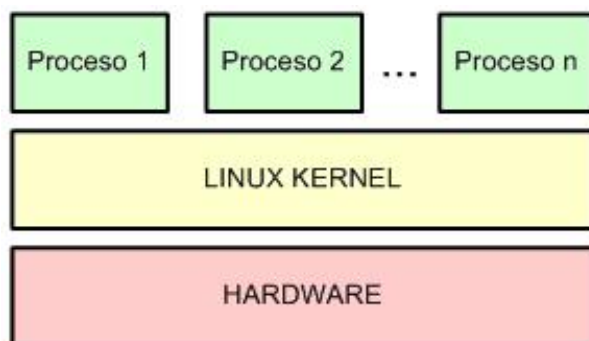


Figura 2. 1 Un típico sistema bajo Linux

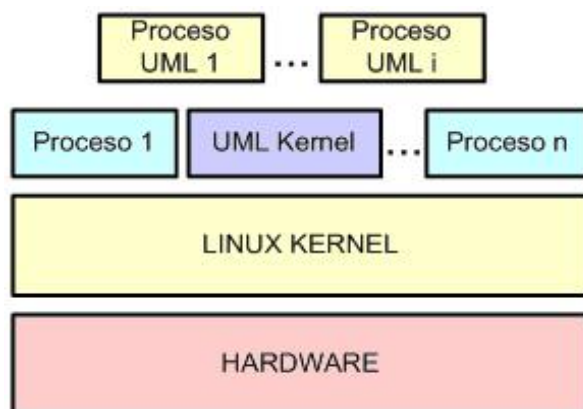


Figura 2. 2 Sistema Linux ejecutando *User Mode Linux*

El *kernel* UML no se comunica directamente con el *hardware*, lo hace a través del *kernel* de Linux de la máquina anfitriona como cualquier otro proceso. Los procesos UML funcionan de la misma manera que en una máquina real. Cuando un proceso dentro de UML realiza una llamada al sistema ésta es interceptada por el *kernel* UML y ejecutada dentro del *kernel* virtual.

UML no accede directamente al *hardware* (tarjeta de red, controlador de disco duro, etc), sino que lo hace a través de la máquina anfitriona.

UML es de gran utilidad cuando se quiere experimentar con nuevas versiones de *software* o realizar modificaciones sobre el código fuente del *kernel*. Una operación no válida o cualquier error involuntario de programación no repercute sobre el sistema operativo anfitrión.

UML permite ejecutar múltiples instancias de Linux utilizando el soporte del sistema operativo anfitrión. De manera análoga a VMware soporta esta característica aunque de un modo más general, puesto que las instancias virtuales no están restringidas a un solo sistema operativo.

### 2.2.1.1 Ventajas de UML

UML es un módulo especial del *kernel* que permite ejecutar varias sesiones de Linux (virtuales) a la vez y fue desarrollado por Jeff Dike[1]. Es un *software* relativamente nuevo con mucho potencial aún inexplorado. Algunas ventajas derivadas del uso de UML son:

- Es libre y presenta el código abierto.
- Su tamaño es reducido y necesita de pocos recursos para poder funcionar. UML no necesita usar X (entorno gráfico). Incluso puede ejecutar varios sistemas con poca memoria.
- Posibilidad de crear varias redes virtuales e incluso crear *routers* virtuales dentro de la red virtual original.
- Soporta *bridging* y *routing*.
- Producción de calidad (permite realizar pruebas experimentales sobre la máquina virtual sin correr riesgos de perjudicar a la máquina real).
- Una arquitectura Linux “oficial”.
- Existe una gran variedad de sistemas de ficheros preconfigurados que pueden ser descargados para utilizarse con UML. Estas imágenes de sistemas de archivos se pueden montar.
- Buen manejo del sistema de archivos, archivos compartidos o archivos .cow

### 2.2.1.2 Desventajas de UML

A continuación se presentan algunas desventajas *User Mode Linux*:

- Actualmente, *User Mode Linux* sólo soporta la creación de Máquinas Virtuales en ambientes Linux, sin embargo, está en etapa de desarrollo una versión que extiende el alcance a sistemas operativos de Microsoft Windows.
- Como es un *software* relativamente nuevo, presenta algunas deficiencias en lo que se refiere a la documentación disponible.
- No hay una interfaz GUI, actualmente toda la configuración e implementación se realiza en base a línea de comandos.
- No hay soporte oficial ni comercial, por tratarse de una herramienta *Open Source*.
- Al igual que VMware, puede ser posible identificar una *Honeynet*<sup>2</sup> UML debido al programa virtual. Sin embargo, el responsable Jeff Dike está tomando medidas para reducir ese riesgo.

---

<sup>2</sup> Máquinas expuestas a Internet para que los *hackers* jueguen con ellas.

## 2.2.2 VMware

Con más de un millón de usuarios registrados y más de cinco mil usuarios empresariales VMware Inc. [VMware] es la compañía líder en el segmento de desarrollo de *software* de virtualización para equipos Intel. Todos sus productos son emuladores de PC; y su diseño obedece a la construcción de una CPU virtual (*software*) que implementa el *set* de instrucciones de la arquitectura X86. Ejecutar instrucciones nativas del *hardware* de una arquitectura de máquina anfitriona tiene su costo, es por ello que el número de máquinas virtuales que se pueden ejecutar es un parámetro acotado. VMware permite crear máquinas virtuales en múltiples plataformas de sistemas operativos [7].

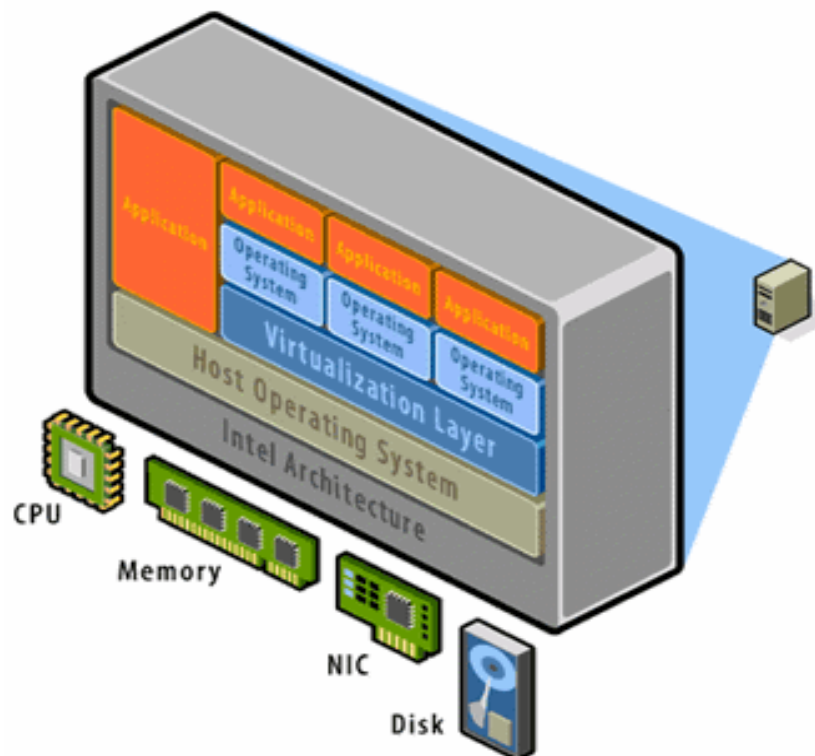


Figura 2. 3: Arquitectura INTEL ejecutando VMware

Inicialmente en el sistema operativo anfitrión se instala VMware y luego se lo ejecuta para iniciar las máquinas virtuales. Es posible ejecutar simultáneamente máquinas virtuales basadas en los distintos sistemas operativos que este producto soporta.

### 2.2.2.1 Ventajas de VMware

VMware es una opción de virtualización muy usada y algunas de las ventajas inherentes a este *software* de virtualización son:

- Soporta una amplia gama de sistemas operativos (Linux, Solaris, Windows y FreeBSD)
- Opciones de red: provee distintos dispositivos para la creación de topologías virtuales de red.

- VMware crea un archivo imagen de cada Sistema Operativo invitado. Ello posibilita que un archivo imagen puede ser migrado a una PC distinta y allí puede ser ejecutado por VMware.
- Fácil de usar. VMware posee una interfaz gráfica que facilita su instalación, y posteriormente la ejecución de las distintas máquinas virtuales.
- Al ser un producto comercial, VMware ofrece soporte técnico y actualizaciones online (parches) a los usuarios registrados.
- Es un producto de muchos años de desarrollo e investigación y por lo tanto altamente estable.

### 2.2.2.2 Desventajas de VMware

En el párrafo anterior se destacaron algunas de las ventajas de VMware, a continuación se listarán las desventajas relacionadas con este *software* de virtualización

- Elevado costo.
- En cuanto a requerimientos del sistema, VMware debe ejecutarse en un entorno X y cada máquina virtual necesitará su propia ventana. Se debe contar con memoria suficiente para el sistema operativo invitado junto al entorno X asociado a cada máquina virtual.
- Cantidad limitada de Sistemas Operativos invitados: VMware sólo puede ejecutar un número pequeño de Máquinas Virtuales (hasta 4).
- Al ser un producto comercial no ofrece el código fuente.
- En lo relativo a seguridad, puede ser posible reconocer VMware en una *honeypot* y por este motivo VMware tiene opciones de configuración que pueden hacer el reconocimiento más difícil (por ejemplo: establecer direcciones MAC para las interfaces virtuales).

### 2.2.3 BOCHS

Es un emulador de PC's *Open Source* escrito en C++. Emula la arquitectura Intel x86 (386, 486, Pentium y Pentium Pro). Es posible ejecutarlo en Linux, Windows 95, Windows NT y DOS [8]. El uso más típico es para la emulación de arquitecturas x86, incluyendo el procesador x86, dispositivos de *hardware* y memoria. Permite simular un monitor sobre la PC. No genera emulación a alta velocidad, para ello debería utilizarse PLEX86 o VMware.

La lentitud hace que el Bochs sea prácticamente inútil para uso continuo. Su configuración no es difícil y el punto más delicado es generar una imagen de disco rígido y configurar los parámetros. También es necesario ajustar manualmente parámetros.

Como característica distintiva, Bochs no utiliza una partición del disco para instalar el

sistema operativo, utiliza un archivo imagen del disco. Bochs soporta dispositivos de CDROM, sonido y tarjetas de red. Algunas aplicaciones que pueden ejecutarse sobre BOSCHS son: Excel 97, Lotus 1-2-3 97, Word 97, PowerPoint 97, Quattro Pro 7 y WordPerfect 7.

Finalmente, BOCHS emula específicamente la arquitectura de un PC IBM/Intel y en este sentido es el más “puro” de todos los emuladores, pues implementa todos los aspectos de la CPU y de la arquitectura en C++. Ejecuta los programas basados en la arquitectura x86 instrucción por instrucción, absolutamente ningún atajo es utilizado a fin de replicar el comportamiento verdadero de un sistema basado en la arquitectura descrita.

## 2.2.4 PLEX86

La contrapartida libre de VMware no es Bochs, sino Plex86, que aún no ha madurado en su desarrollo pese a tener varios años de evolución. Plex86 tiene características similares a VMware, y a diferencia de este sólo se ejecuta en ambientes Linux. Al emular el set de instrucciones de CPU's que responden a las arquitecturas X86, extrae las ventajas de la excelente habilidad que posee Linux para comunicarse con el *hardware* [12].

Es importante destacar que Plex86 fue concebido como una alternativa *Open Source* a VMware, pero las expectativas iniciales nunca se concretaron y el proyecto sufrió continuas detenciones y postergaciones. A pesar de ello, PLEX86 fue una obra de inspiración para otros desarrollos en lo que a herramientas de virtualización se refiere.

## 2.2.5 QEMU

Es un emulador de procesador que utiliza traslación dinámica de archivos a buena velocidad. QEMU no requiere que se *patchee* el *kernel* del *host* en que se ejecuta y por lo tanto es seguro y fácil de usar [10].

Soporta prácticamente casi todos los sistemas operativos y puede resultar muy interesante para ejecutar distintos sistemas operativos sobre Linux. A pesar de que la filosofía es básicamente la misma del Bochs, el QEMU consigue una *performance* mejor a través de un proceso de compilación de las instrucciones de la CPU emulada. La compilación es lenta, pero el resultado es almacenado en un *cache*, lo que diluye el costo de compilación si el mismo código es ejecutado varias veces. Bochs es un emulador más eficiente sin embargo para las aplicaciones practicas es más conveniente utilizar QEMU. Una gran ventaja del QEMU sobre el Bochs es que este se encuentra preparado para emular varios procesadores y arquitecturas.

QEMU tiene dos modos de operar:

- Emulación Completa: emula un sistema completo, por ejemplo, un PC con un procesador y varios periféricos. Este modo puede ser usado para iniciar diferentes sistemas operativos.
- Emulación del lado del usuario (solamente en un *host* con linux): el QEMU tiene un modo *user-level*, sólo para Linux. En este modo, la emulación es hecha en el contexto de un ejecutable Linux. Sólo la CPU y la memoria virtual son emuladas y no el *hardware*. Se dice que la velocidad percibida de

programas típicos puede llegar al 90% de la versión nativa.

Una propiedad opcional del QEMU es el módulo acelerador que permite optimizar una PC que esta siendo emulada sobre otra PC. Este módulo permite ejecutar el código de las aplicaciones directamente sobre el procesador del *host* alcanzando una *performance* cercana a la nativa. Esto exige la instalación de un módulo en el *kernel* del sistema anfitrión. El uso de este recurso dobla la velocidad de emulación, pero quiebra el aislamiento completo del sistema emulado.

En la máquina emulada por el QEMU, todos los periféricos importantes, inclusive la placa de red son PCI. Ello facilita las cosas con sistemas operacionales recientes. Linux ejecutado dentro del QEMU es rápido en el modo texto, pero muy lento (peor que Windows XP) en el modo gráfico.

## 2.2.6 WINE

El proyecto WINE es bastante antiguo, ya que nació en 1993. En aquel entonces el objetivo del mismo era lograr ejecutar programas de Windows 3.1 en GNU/Linux. Como su nombre lo indica, WINE no es un emulador. La idea detrás de este proyecto es la de clonar la Win32 API (una gran cantidad de características que facilitan la creación de *software* bajo Windows).

Para entender de una manera más clara el funcionamiento de este *software* hay que imaginarlo como una capa de compatibilidad con Windows. WINE provee lo siguiente:

- Un conjunto de herramientas de desarrollo para portar códigos fuente de aplicaciones Windows a *Unix*.
- Un cargador de programas, el cual permite que muchas aplicaciones para Windows 3.x/9X/ME/NT/2000/XP se ejecuten sin modificarse en varios *UNIX* para plataforma Intel como GNU/Linux, FreeBSD y Solaris.

WINE no requiere que se encuentre instalado Microsoft Windows, dado que es una implementación alternativa que no utiliza código fuente de Microsoft. Pero es posible que utilice alguna biblioteca dinámica en el caso que se encuentre instalado Windows.

Dada la complejidad del objetivo perseguido por este proyecto y a su vez que la meta se encuentra en constante movimiento (cada nueva versión de Windows implica nuevos desarrollos), el proyecto aún no llegó a lanzar la versión 1.0. Cuando se logre desarrollar un sistema de instalación simple e intuitiva para los usuarios hogareños y soportar a las aplicaciones que fueron creadas específicamente para Windows XP se espera que sea lanzada la versión 1.0.

En el sitio *web* del proyecto se puede votar para elegir qué aplicación se desea que pueda ejecutarse en WINE. El *software* más votado es el Visual Fox Pro de Microsoft, luego lo siguen *Macromedia Dreamweaver* y el Microsoft Word. En el caso del procesador de textos de Microsoft, ya se ha podido ejecutar bajo WINE, pero su *performance* aún deja mucho que desear. También hay otro sistema, el cual se utiliza para calificar el funcionamiento de las aplicaciones bajo WINE. Esto permite que los usuarios califiquen qué nivel de funcionalidad obtuvieron del *software* al ejecutarlo bajo WINE. El código fuente de WINE se encuentra licenciado bajo la LGPL por lo que se lo considera *Software Libre*.

### 2.2.6.1 Ventajas de WINE

Algunas de las ventajas derivadas de la utilización de WINE como herramienta de virtualización son:

- Ejecuta aplicaciones basadas en Windows eficazmente bajo el sistema operativo Linux.
- Es posible asignar un alto grado de prioridad a un proceso basado en Windows a fin de aumentar el desempeño del programa.
- No se requiere de una copia del sistema operativo Windows.

### 2.2.6.2 Desventajas de WINE

Algunas de las desventajas derivadas de la utilización de WINE son:

- No se puede ejecutar todo el universo de aplicaciones Windows en Linux.
- La evolución de WINE es lenta y dificulta la ejecución de aplicaciones recientes basadas en nuevas versiones del sistema operativo Windows.

## 2.2.7 DOSBOX

DOSBox posee una filosofía de diseño preparada para emular una parte del set de instrucciones de distintas arquitecturas de microprocesadores. Sus mentores tuvieron como premisa de diseño la ejecución de juegos basados en MS-DOS en distintas plataformas de sistemas operativos tales como: Windows, Linux, MacOS X, etc.

Una gran ventaja del DOSBox sobre el Bochs es la facilidad de uso. No es necesario crear imágenes de disco, ni instalar un sistema operacional. El DOSBox viene con una mini-imagen de disco. Es posible visitar directorios del sistema anfitrión “montándolos” dentro del emulador.

En una prueba de compilación de programa MS\_DOS, el DOSBox mostró ser más rápido que el Bochs, aunque más lento que el Qemu. Se debe tomar cuidado al usar el DOSBox para cualquier cosa no relacionada a su perfil original, cuando esta es la necesidad existen herramientas tal como DOSEMU que constituyen apuestas más seguras. En el caso de aplicaciones relacionadas a juegos, el DOSBox es perfecto y altamente recomendable.

## 2.2.8 DOSEMU

Es un intermedio entre Linux y FreeDos<sup>3</sup>, o algún otro sistema DOS, el cual permite que se

---

<sup>3</sup>FreeDOS es libre y compatible con el sistema operativo MS-DOS. Es un proyecto que comenzó en 1994. La herramienta Dosemu ejecuta el *kernel* de FreeDos para emular el DOS.



ejecuten programas en DOS bajo linux, ambos desde la terminal [13]. DOSEMU tiene prácticamente la misma filosofía del VMware, sin embargo es un *software* libre.

DOSEMU suministra los servicios de la BIOS, y también emula algunos dispositivos de *hardware* del IBM, inclusive placa de sonido y vídeo VGA. La velocidad de los programas en este ambiente es bastante buena. El soporte a programas "serios" es bastante bueno. El soporte a juegos es bajo (tienden a no funcionar o a funcionar de forma errática). A pesar del nombre, el DOSEMU no sustituye el MS-DOS y por sus características DOSEMU sólo funciona en ambientes Linux/Intel.

### 2.2.9 WIN4LIN

Es un emulador que se ejecuta sobre Linux, requiere una licencia de Windows. Actualmente emula Windows 98 y Windows ME [15]. El soporte para Windows XP está siendo desarrollado, aunque soporta más aplicaciones de Windows que WINE y tiene mejor *performance* que VMware.

## 2.3 Selección de la Herramienta de Virtualización

En el punto anterior se enumeraron distintos *software's* de virtualización disponibles en el mercado, por lo que en esta sección resta elegir el *software* de virtualización a utilizar. La selección se realizará tomando en cuenta las ventajas y desventajas de cada *software* así como también los aspectos básicos que debe cubrir dicho *software* para el proyecto en particular. Es importante destacar que mediante una investigación minuciosa se llegó a la conclusión de que la mayor parte de los laboratorios virtuales en el área considerada (Redes) están basados en UML como Adios[28], VNUML[29], Netkit[30], Virtual Network Laboratory(CPIT)[32]. Las ventajas de UML sobre otras herramientas de virtualización fueron descritas en la sección 2.2.1. Por todo lo hasta aquí expresado *User Mode Linux* fue seleccionado como herramienta de virtualización. Algunos de los aspectos por los cuales sobresalió UML sobre el resto de las herramientas de virtualización consideradas son los siguientes:

- *Software Open Source* de fácil uso y configuración.
- Posee un gran potencial todavía inexplorado.
- Cuenta con las características apropiadas para practicar y entender mejor las tecnologías que disponemos.
- Es de reducido tamaño y se necesitan pocos recursos de *hardware* en comparación con otras herramientas.
- Posee todas las herramientas para crear dispositivos de red virtuales (*hub's*, *switch's* y *router's*)