

Montevideo, diciembre de 2004

Universidad ORT Uruguay

Administración de Servidores Linux.

Máquinas virtuales y Emuladores.
Cómo ejecutar aplicaciones de otros sistemas operativos en
Linux.

Autores: Felipe Vázquez y Diego Invernizzi

Docente: Enrique Verdes

Abstract

El presente documento ofrece una introducción al lector a las máquinas virtuales y emuladores. Partiendo de la definición de los conceptos hasta llegar a aplicaciones reales de máquinas virtuales y emuladores brindando ejemplos prácticos. Por último se comparan algunos productos de referencia en el mercado actual, analizando sus puntos fuertes y débiles.

Índice

Introducción.....	
Que son las Máquinas virtuales.....	
Que son los Emuladores.....	
Programas de Máquinas Virtuales.....	
BOCHS.....	
VMWare.....	
Plex86.....	
PearPC.....	
QEmu.....	
Java Virtual Machine.....	
Programas de Emuladores.....	
WINE.....	
CrossOver Office.....	
Emuladores de Consolas de Videojuegos y Arcades.....	
Emuladores de PCs obsoletos.....	
Conclusiones.....	

Introducción

En el ámbito informático, la posibilidad de disponer de un entorno de pruebas en donde ensayar soluciones o adquirir conocimientos sobre productos o tecnologías, se convierte en el sueño recurrente de todo administrador de sistemas. Incluso, en aquellos casos en los que si se cuenta con un entorno de testeo o laboratorio, el espacio creado no suele ser perfecto. Para estos menesteres, lo habitual suele ser dedicar equipos obsoletos que poco o nada tienen que ver con el entorno de producción real con lo que hay que tener en cuenta tantas excepciones en los test y prácticas, que los resultados acaban desvirtuados. En estos casos la solución llega de la mano de programas de máquinas virtuales.

Muchas impresoras, por ejemplo, están diseñadas para emular a las Hewlett-Packard LaserJet porque mucho software esta escrito para impresoras HP. Emulando una impresora HP, una impresora puede trabajar con cualquier software escrito para una impresora HP real. La emulación engaña al software haciéndolo creer que un cierto aparato es realmente otro aparato. Los paquetes de software de comunicaciones a menudo incluyen controladores emulados de terminales. Esto permite a tu PC emular un tipo particular de terminal para que puedas conectarte a una mainframe. También es posible que una computadora emule otro tipo de computadora.

Un emulador es un programa o dispositivo que pretende ser otro programa o dispositivo en particular necesario para que otros componentes puedan funcionar, para lograr esto emula (imita) funciones y características propias de la otra máquina o dispositivo.

Que son las Máquinas virtuales.

Son programas que permiten simular en un mismo equipo y de manera simultanea, varios ordenadores. Cada uno de ellos con su propia configuración, hardware y software, sistema operativo y programas. Es decir, en un mismo ordenador un usuario puede disponer de un ordenador que ejecuta Linux, un ordenador que corre bajo WinXP y hasta un servidor de Windows 2003. Todos ellos funcionando a la vez, sin necesidad de reiniciar el ordenador para pasar de uno a otro. Separados pero interrelacionados entre ellos de tal forma que los cambios que se hagan en uno de ellos, puedan repercutir inmediatamente en el otro. Por ejemplo un programador puede tener en su ordenador todas la herramientas de desarrollo que necesita para crear sus programas y puede realizar mientras desarrolla, pruebas de comportamiento sobre diferentes SO o arquitecturas simplemente cambiando de ventana y ejecutando la aplicación que desea probar.

Los programas de Máquinas virtuales, se encargan de simular el entorno de hardware que puede necesitar un sistema operativo y regular el acceso a los distintos dispositivos que los equipos creados puedan tener. Para ello se conjugan varios elementos. Por un lado el ordenador físico, la máquina anfitrión (Host) que aloja un sistema operativo que permite hacer funcionar esa máquina y sobre el que se ejecuta la máquina virtual. Por otro lado los ordenadores huéspedes (Guest) que son gestionados por el programa de máquina virtual, que quedan definidos al asignarles recursos de hardware simulados e instalando su correspondiente sistema operativo. Los ordenadores huéspedes se ejecutan simultáneamente y pueden compartir por ejemplo una tarjeta de red, porque no acceden directamente al dispositivo físico, el programa de máquina virtual se encarga de presentar un dispositivo ficticio e interceptar las distintas peticiones de uso y gestionarlas., lo mismo ocurre con el resto de los componentes.

Constitución de una máquina virtual o extendida

Los sistemas operativos, ponen al servicio del usuario una *máquina virtual* cuyas características son distintas (y más fáciles de abordar) que las de la máquina real subyacente. Algunas áreas en las que es frecuente que la máquina virtual difiera

de la máquina real que la soporta son:

Entrada/salida (E/S).

La capacidad de E/S de un *hardware* básico puede que sea extremadamente compleja y que requiera sofisticados programas para su utilización. Un sistema operativo evita al usuario el problema de tener que comprender el funcionamiento de este *hardware*, poniendo a su alcance una máquina virtual mucho más sencilla de usar.

Memoria.

Muchos sistemas operativos presentan la imagen de una máquina virtual cuya memoria difiere en tamaño de la de la máquina real subyacente. Así, por ejemplo, un sistema operativo puede emplear memoria (en discos) para crear la ilusión de una memoria principal mucho más extensa de la que se dispone en realidad. Alternativamente, puede repartir la memoria principal entre varios usuarios, de forma que cada uno de ellos "vea" una máquina virtual cuya memoria sea menor que la de la máquina real.

Sistema de ficheros.

La mayoría de las máquinas virtuales incluyen un sistema de ficheros para el almacenamiento a largo plazo tanto de programas como de datos. El sistema de ficheros está basado en la capacidad de almacenamiento sobre cinta o disco de la máquina real. El sistema operativo, sin embargo, permite al usuario acceder a la información almacenada a través de nombres simbólicos en lugar de hacerlo a través de su posición física en el medio de almacenamiento.

Interacción a nivel de programa.

Una máquina virtual puede posibilitar la interacción entre distintos programas de los usuarios de forma que, por ejemplo, la salida de uno de ellos se emplee como entrada de otro.

La naturaleza concreta de una máquina virtual dependerá de la aplicación particular a la que se dedique. Así, por ejemplo, las características de una máquina virtual que controle un sistema de tiempo real serán distintas de las de una máquina virtual que se utilice para el desarrollo de programas.

Que son los Emuladores

Haciendo un poco de historia vemos que una vez más, una tecnología informática que nos parece moderna resulta remontarse a una época bien lejana. De el fin de la segunda guerra mundial a los comienzos de los años 80, es un hecho que IBM (Internacional Business Machines) compañía fundada en 1911, ha reinado en el mundo de la informática. Durante los años 50/60, de aparatos tales como la IBM 1401 hacen figura de referencia absoluta en materia de ordenadores.

Concientes que esta supremacía reposa mas sobre una reputación que sobre las performances reales de las máquinas, los dirigentes de IBM inician entonces un estudio conocido hoy bajo el nombre de SPREAD Report, que tiene por objetivo de conducir a la concepción de un producto totalmente nuevo, que pudiera distanciar durablemente la concurrencia. Uno ve entonces una época donde los ordenadores son concebidos como máquinas totalmente independientes una de las otras. No es cuestión de componentes compartidos ni de compatibilidad. Cada sistema necesita por lo tanto sus propios datos logísticos, su propio hardware adaptado al uso que se va hacer y los responsables de la búsqueda de IBM encuentran una pista a explorar.

En 1962, un entendimiento circula en IBM según el cual un “usuario anónimo” puede modificar el hardware de una IBM 705 a fin que pueda ejecutar programas escritos para la IBM 1401, modelo más antiguo. Este entendimiento va definitivamente a indicar a los científicos de la compañía el camino a seguir. Es la época donde IBM se apresta a lanzar su NPL (New Product Line), serie de ordenadores revolucionarios y fruto de su esfuerzo de desarrollo tecnológico.

Fin de Septiembre del 63, IBM confía a uno de sus laboratorios situados en Francia, la misión de desarrollar una serie de programas de simulación teniendo como objetivo imitar el comportamiento de sus siete ordenadores mas populares y establecer una compatibilidad descendiente entre ellos, todo sobre el plano puramente programador. Los primeros resultados son desalentadores, la simulación tiene un funcionamiento al menos dos veces mas lento que los ordenadores en cuestión, y ella es inutilizable en un marco profesional.

Durante este tiempo, los ingenieros de IBM no se durmieron, Honeywell

saca la H-200, un ordenador capaz de hacer funcionar los programas IBM. Ya no es mas cuestión de compatibilidad descendiente al seno de una línea de ordenadores de una misma marca, más categóricamente de compatibilidad con otros sistemas y otras marcas. John Haanstra, otro ingeniero de IBM, propone una solución hardware, que va a resultar una IBM 1401 S, una máquina que permite a IBM recuperar temporalmente su retraso tecnológico sobre Honeywell, mas no prefigura ninguna línea de ordenadores compatibles entre ellos y se destaca entre los otros sistemas gracias a su flexibilidad programadora.

Larry Moss, un joven talento muy prometedor, lanza la teoría de que un ordenador puede ser compatible con otro, el hardware de las dos máquinas debe ser lo mas similar posible, permitiendo a la vez la ejecución de programas para los cuales ellos fueron concebidos y el lanzamiento de un programa de simulación como en el que el equipo francés ha trabajado y se vuelca por lo tanto en el estudio de una solución de compatibilidad mezclando el entendimiento material y logístico. Ellos consiguen poner a punto una extensión para los ordenadores NPL haciéndolos capaces de ejecutar programas previstos para IBM7070, el mas sofisticado de los ordenadores de IBM de entonces. Los trabajos de Moss dan como resultado un funcionamiento perfecto de estos programas, no mostrando ninguna perdida de performance. El decide mostrar la superioridad de su proyecto desarrollado sobre la “simulación”, de conferirle la denominación de “emulación”, expresión que indicaba una corrección, un progreso.

En Abril de 1964 ve la luz la familia de ordenadores IBM: System / 360, y el emulador Moss, conocido también con el nombre de 7070 Emulator, donde Joe Brown y Larry Moss son los padres. La aparición va a conocer un franco suceso, los poseedores de la System /360 estaban muy contentos de poder continuar utilizando los viejos programas para la 7070 y su comercialización durara hasta el debut de los años 70. La continuación de la historia no es mas que una larga serie de proyectos de emuladores utilizando o no extensiones materiales, y haciendo potencialmente compatibles máquinas mas o menos diferentes unas de las otras.

Hoy el término ha reencontrado su primer significado, como una verdadera carrera que se libran los pequeños genios de la programación. Nuevos emuladores salen cada mes y algunas máquinas a emular, o sobres las cuales emular, los especialistas se vuelcan hacia los ordenadores, juegos de arcade o consolas que son todavía de actualidad y la emulación deviene en piratería.

Un emulador es un programa o dispositivo que pretende ser otro programa o dispositivo en particular necesario para que otros componentes puedan funcionar, para lograr esto emula (imita) funciones y características propias de la otra máquina o dispositivo. Se desarrolla un emulador cuando un hardware particular ya no está disponible en el mercado pero algunas de las aplicaciones que corrían sobre él siguen siendo utilizadas y necesitan de ese hardware. La práctica de utilizar un emulador para hacer funcionar antiguos programas sobre nuevos dispositivos es llamada "Emulación de Terminales".

Pero los emuladores no se utilizan solamente con programas antiguos. Por ejemplo, si hablamos de un emulador de Linux, nos referimos a un programa que imita a ese sistema operativo, reconociendo ciertos comandos, acciones pero funcionando desde otro sistema operativo, como Windows. También tenemos un buen ejemplo en los emuladores de consolas, programas que permiten jugar a juegos típicos de consolas con el interface de las mismas desde el ordenador. Otro ejemplo, son los programas que permiten que una Apple Macintosh emule a una PC.

También es posible para una computadora, emular otro tipo de computadora, por ejemplo para ejecutar programas desarrollados para PCs en un Apple Macintosh

Es importante comprender que emulación no es lo mismo que simulación. La emulación es un intento de imitar el diseño interno de un dispositivo. Simulación es un intento de imitar las funciones de un dispositivo. Por ejemplo, un programa imitando el hardware del arcade Pacman y ejecutando la ROM real de Pacman en él es un emulador. Un juego de Pacman escrito para tu computadora pero usando gráficos similares a los del arcade real es un simulador

Una forma interesante de comprender que es un emulador, es ver que puede ser emulado. Básicamente, cualquier cosa que tenga un microprocesador dentro puede ser emulada. Por supuesto, sólo los dispositivos ejecutando programas más o menos flexibles son interesantes para emular. Esto incluye:

- Computadoras
- Calculadoras
- Consolas de VideoJuegos

- VídeoJuegos Arcade

Hay tres esquemas básicos que pueden ser usados para un emulador. Pueden ser combinados para obtener el mejor resultado.

1) Interpretación

Un emulador lee código emulado byte a byte desde la memoria, lo decodifica y realiza las operaciones apropiadas en los registros emulados, memoria y E/S. Las virtudes de este modelo incluyen facilidad de depuración, portabilidad y facilidad de sincronización. Una debilidad simple, grande y obvia es el bajo rendimiento. La interpretación toma mucho tiempo de CPU y puedes necesitar una computadora bastante rápida para ejecutar el código a una velocidad decente.

2) Recopilación estática

En esta técnica se toma un programa escrito en el código emulado y se intenta traducir éste al código ensamblador de tu computadora. El resultado será un ejecutable que puedes hacer funcionar en tu computadora sin ninguna herramienta especial. Aunque esto suena muy bien, no siempre es posible. Por ejemplo, no puedes recompilar estáticamente código que se automodifique a él mismo ya que no hay manera de saber a qué hay que convertirlo sin ejecutarlo. Para evitar ese tipo de situaciones, es posible combinar la recopilación estática con un intérprete o recompilador dinámico.

3) Recopilación dinámica

La recopilación dinámica es esencialmente la misma cosa que la estática, pero ocurre durante la ejecución del programa. En lugar de intentar recompilar todo el código de una vez, se hace al vuelo cuando se encuentra una instrucción CALL o JUMP. Para incrementar la velocidad, esta técnica puede ser combinada con la recopilación estática.

Programas de Máquinas Virtuales

Las máquinas virtuales mas difundidas en sistemas Linux son:

- **BOCHS**
- **VMWare**
- **Plex86**
- **PearPC**
- **QEmu**
- **Java Virtual Machine**

BOCHS

BOCHS es un programa bajo licencia GPL, programado en c++ para simular o emular un ordenador donde instalar un sistema operativo y poder ejecutar aplicaciones en él.

Emula un PC de tipo 386, 486, Pentium, Pentium Pro o AMD64. Incluye opciones MMX, SSE, SSE2 y también instrucciones 3DNow. Puede emular la CPU, la BIOS y cargar la imagen de cualquier sistema operativo del tipo Windows o Linux. Es decir, podremos por ejemplo ejecutar un Sistema Windows 2000 desde una ventana de un Sistema Linux o viceversa.

A diferencia de otros más populares, su interfaz gráfica de usuario es menos intuitiva pero ofrece una interesante personalización. A su vez, y a diferencia de VMWare, Bochs emula un sistema X86 independientemente de la plataforma en la que esta corriendo, es decir que no depende de las instrucciones nativas del procesador en el que corre. Esto le da a este software tanto una fortaleza como una debilidad.

Fortaleza porque permite crear una maquina virtual arquitectura X86, por ejemplo dentro de una estación de trabajo SUN o Alpha.

Debilidad, porque para lograr lo anterior, emplea simulación por software de cada instrucción X86 generada, lo que implica una baja en la performance del Sistema virtual.

Otros software comerciales como VMWare pueden lograr una emulación mucho más rápida usando una técnica llamada virtualización⁽¹⁾ pero tienen dos grandes contras:

- No tienen la portabilidad a arquitecturas no X86
- No son Open source

Hay un proyecto open source que se está desarrollando que utiliza la tecnología de virtualización llamado Plex86 que mencionaremos más adelante.

⁽¹⁾ La virtualización es una técnica que toma ventajas de la emulación de instrucciones X86 en sistemas X86, permitiendo que grandes porciones de la simulación se realicen a velocidad nativa del hardware.

Las versiones de este emulador disponibles son para las siguientes plataformas (Sistemas Host):

- Familia Windows
- Familia Unix (Linux, BSD, Minix)
- Sistemas Mac
- Solaris
- BeOS

Y en ellos se pueden emular los siguientes sistemas Operativos (Sistemas operativos virtuales o huéspedes)

- Familia Windows: Windows 3.1, 3.11, W95, W98, W2K WXP
- Familia Unix (Linux, BSD, Minix)
- Sistemas Mac
- Solaris
- BeOS

Tipo de Licencia: LGPL
Lenguaje Programación: C++
Ultima Versión 2.1.1
Estado de Desarrollo: 4 (Beta)
Interface de usuario: X Windows System (X11)
Pagina Oficial: <http://bochs.sourceforge.net/>

Historia:

- Escrito por Kevin Lawton en 1994
- Paso a ser un software comercial hasta marzo 2000, año en que lo compra MandrakeSoft y lo hace Open Source bajo licencia GNU LGPL.
- En marzo del 2001 se muda el proyecto de bochs.com a bochs.sourceforge.net
- Es elegido el Proyecto del mes por Sourceforge en Agosto 2004.

VMWare

VMWare es una alternativa comercial a Bochs para emular o simular diferentes sistemas operativos. No es multiplataforma, funciona en sistemas con arquitectura x86 tales como Windows, Linux, DOS, BSD, etc.

Puede emular por ejemplo tarjetas gráficas o tarjetas de red. Y además también se puede acceder al hardware del ordenador, como a dispositivos USB, unidades de disco, puertos de entrada, etc.

La creación de un sistema virtual es realmente sencilla, ya que permite configurar muchos aspectos del Hardware de la maquina virtual como ser el tamaño de las unidades de disco, la cantidad de memoria RAM del sistema virtual o la cantidad y tipo de los demás componentes emulados.

Además, el hardware emulado, independientemente del hardware de la maquina host, es muy conocido y sencillo de instalar, no encontrando en las 3 instalaciones que hemos realizado (Win95, Win98, Knoppix 3.6) problemas de configuración.

Puede compartir la misma tarjeta de red en todos los sistemas operativos a la vez, incluido el sistema base, cada uno con su propia IP y MAC. Es muy fácil emular una LAN con este software ya que alcanza con que los Sistemas virtuales y

el sistema propietario se encuentren en el mismo dominio.

Dentro de las características mas sobresalientes de esta maquina virtual debemos destacar la velocidad de la emulación y la posibilidad de poner en pausa un sistema virtual (Guest) no teniendo necesidad de arrancarlo y apagarlo cada vez. Este emulador maneja a su vez el concepto de SnapShot o sea foto del sistema que me permite sacarle una foto al sistema emulado en un punto dado para poder iniciarlo desde ese punto cuantas veces quiera.

Especificaciones del Hardware Emulado

Procesador:

- Intel Pentium II o posterior, o AMD Athlon o posterior, dependiendo del procesador de la maquina host.
- Soporte experimental para procesadores AMD64 Opteron, Athlon 64 e Intel IA-32e

Memoria:

- Hasta 3.6GB, dependiendo de la memoria del Sistema Propietario
- Máximo de 4GB total disponible para todas las máquinas virtuales

Discos IDE:

- Hasta 4 dispositivos IDE
- Los discos pueden ser virtuales o dispositivos físicos
- Admite emulación de CD-ROM y DVD-ROM
- Los discos IDE virtuales pueden ser de hasta 128GB

CD Drives:

- CD-ROM (CD-Recordable o CD-Rewritable drive) o DVD-ROM drives basados en discos IDE o SCSI del sistema propietario.
- CD-ROM pueden ser dispositivos físicos o imágenes ISO CD-ROM

Dispositivos SCSI:

- Hasta siete dispositivos
- Los discos pueden ser virtuales o dispositivos físicos
- Los discos IDE virtuales pueden ser de hasta 256GB

- El soporte SCSI genérico admite el uso de dispositivos sin necesidad de sus correspondientes drivers en el SO host
- Trabaja con scanners, CD-ROM, DVD-ROM, tape drives, y otros dispositivos SCSI
- Emula el hardware LSI Logic LSI53C1030 Ultra320 SCSI I/O controller
- Emula el hardware Mylex (BusLogic) BT-958

Gráficos:

- VGA and SVGA con soporte VESA BIOS

Floppy Drives

- Hasta dos disqueteras 1.44MB

Serial (COM) Ports

- Hasta cuatro puertos seriales (COM ports)
- Salida a puerto serial o a archivos en la maquina host

USB Ports

- dos puertos USB 1.1 UHCI controller
- soporte de dispositivos USB incluyendo impresoras, escaners, PDAs, HDD, memory card readers, y cámaras digitales.

Parallel (LPT) Ports

- hasta dos puertos bi-direccionales (LPT)

Teclados y dispositivos de puntero

- teclado extendido de 104 teclas
- mouse serie PS/2
- Soporte p/ Serial tablets

Tarjetas de Red

- Hasta tres tarjetas virtuales Ethernet
- AMD PCnet-PCI II compatible

Sonido

- Sonido entrante y saliente emulando Creative Sound Blaster Audio PCI

compatible (MIDI sound and joysticks not supported)

BIOS

- PhoenixBIOS 4.0 Release 6-based VESA BIOS with DMI version 2.2/SMBIOS support

Virtual Networking System

- Nueve virtual Ethernet switches
- dispositivo Virtual NAT
- servidor Virtual DHCP
- adaptador de red virtual
- Virtual network bridge con soporte de red alámbrico e inalámbrico
- Virtual networking system con soporte para protocolos TCP/IP, Microsoft Networking, NetBEUI, Samba, NetWare, y NFS
- Built-in NAT supports client software using TCP/IP, FTP, DNS, HTTP, Telnet, and NetLogon

Requerimientos del Sistema Host:

Hardware PC:

PC estandard; 500MHz o mas compatible x86 (recomendado; 400MHz mínimo)

Procesadores compatibles:

- Intel: Celeron, Pentium II, Pentium III, Pentium 4, Pentium M (incluido Centrino mobile technology), Xeon (incluido "Prestonia")
- AMD: - Athlon, Athlon MP, Athlon XP, Duron, Opteron

Memoria:

Memoria suficiente para correr el SO propietario más la memoria requerida para cada uno de los sistemas virtuales y para las aplicaciones de todos estos.

Se recomiendan por lo menos 256MB aunque se puede ejecutar con 128MB.

Sistemas Operativos Hosts soportados

Windows Host Operating Systems

Windows Server 2003 Web Edition

Windows Server 2003 Standard Edition

Windows Server 2003 Enterprise Edition

Windows XP Professional and Windows XP Home Edition c SP1 o 2

Windows 2000 Professional Service Pack 3 or 4

Windows 2000 Server Service Pack 3 or 4

Windows 2000 Advanced Server Service Pack 3 or 4

Windows NT Workstation 4.0 Service Pack 6a,

Windows NT Server 4.0 Service Pack 6a,

Windows NT 4.0 Terminal Server Edition Service Pack 6.

Linux Host Operating Systems

Mandrake Linux 9.0 - 8.2

Red Hat Enterprise Linux 3.0 - 2.1

Red Hat Linux Advanced Server 2.1

Red Hat Linux 9.0 - 8.0 - 7.3 - 7.2 - 7.1 - 7.0

SuSE Linux 9.1 - 9.0 - 8.2 - 8.1 - 8.0 - 7.3

SuSE Linux Enterprise Server 8 - 7

Tipo de Licencia:	Comercial U\$\$189 distribución electronica U\$\$199 distribución en CD
S.O. soportados:	All POSIX (Linux/BSD/UNIX-like OSes), Linux Familia Windows
Ultima Versión	VMWare Workstation 4.5.2
Estado de Desarrollo:	5 - final
Interfase de usuario:	X Windows System (X11)
Pagina Oficial:	http://www.vmware.com/

Plex86

Plex86 es una utilidad de Linux que emula el comportamiento de un PC dentro del propio sistema operativo, al igual que hace VMWare, pero con licencia abierta y con la restricción, ahora, de que el sistema emulado puede ser solamente Linux/x86.

De manera que fácilmente podemos ejecutar varios sistemas operativos Linux de forma concurrente sobre un sólo PC basado en Linux. La primera utilidad que podemos darle a Plex86 es la de tener un PC configurado con varias distribuciones linux para prueba o soporte.

Es bastante mas rápido que Bochs ya que utiliza al igual que vmware, virtualización pero a diferencia de este último es de licencia LGPL.

Puede ser utilizado en conjunción con Bochs como un acelerador para ejecutar SO no VM-amigables.

Tipo de Licencia:	GNU Library o Lesser General Public License(LGPL)
S.O. soportados:	All POSIX (Linux/BSD/UNIX-like OSes), Linux
Lenguaje Programación:	C
Estado de Desarrollo:	4 (Beta)
Pagina Oficial:	http://plex86.sourceforge.net/

PearPC

Es una maquina virtual de arquitectura PowerPC para x86 capaz de ejecutar la mayoría de los Sistemas Operativos para PowerPC. Existen versiones de esta MV para Windows y Linux

Sistemas Operativos Soportados:

Host	POSIX-X11 (Linux, ...), Win32
Guest:	Mandrake Linux 9.1 p/PPC Darwin p/PPC Mac OS X 10.3 OpenBSD for PPC (en desarrollo) NetBSD for PPC (en desarrollo) AIX for PPC (See FAQ.)

Limitaciones

La emulación de CPU es lenta, de alrededor de 15 veces a 500 veces mas lenta de acuerdo al procesador de la maquina host.

La emulación de de los HDD y CDRoms es bastante rápida especialmente en SO Host que soporten Bus-mastering (Linux, Darwin, Mac OS X).

No se ha implementado aún la posibilidad de guardar y recuperar estados en la maquina virtual.

No hay soporte para discos mayores a 128 GiB.

Esta en plena etapa de desarrollo, habría que ver este proyecto dentro de un tiempo para comprobar sus avances.

Tipo de Licencia:	General Public License(GPL)
Lenguaje Programación:	C++, C and (on x86 platforms) assembler
Estado de Desarrollo:	3 (Alpha)

Operating System: All 32-bit MS Windows (95/98/NT/2000/XP),
All POSIX (Linux/BSD/UNIX-like OSes), BeOS

Última Versión: 0.3.1 – 5/09/2004

Página Oficial: <http://pearpc.sourceforge.net>

User Interface: Win32 (MS Windows), X Window System (X11)

QEmu

Es una Maquina virtual libre y gratuita que nos permite emular arquitecturas de procesadores tales como: x86, Sparc, ARM y PowerPC. Con él podemos ejecutar Sistemas Operativos para estas arquitecturas en una ventana aparte en nuestro escritorio, es decir, podemos tener dentro de nuestra distribución Linux de plataforma x86 una ventana aparte en la que corra un Mac OS X en un PowerPC.

Actualmente el emulador corre bajo Linux y Mac OS X, pero se está desarrollando una versión para Windows.

Características:

QEMU es un emulador de procesadores bastante rápido, que posee dos modos de operación:

Full system emulation:

En este modo, QEMU emula un sistema completo (por ejemplo un PC), incluyendo un procesador y varios dispositivos. Puede ser usado para ejecutar diferentes sistemas operativos sin necesidad de rebootear la maquina.

User mode emulation (solamente para hosts Linux):

En este modo, QEMU puede lanzar procesos linux compilados para un procesador en otro procesador. Puede ser usado para lanzar el emulador de las API de windows Wine en una arquitectura no x86.

CPU emuladas soportadas:

Target CPU	User emulation	System emulation
X86	OK	OK
ARM	OK	Not supported

SPARC	Testing	Testing
PowerPC	OK	Testing

Host CPU soportadas

Las *host* CPU son aquellas en la que se ejecuta QEMU.

Host CPU	Status
X86	OK
PowerPC	OK
AMD64	Testing
Alpha	Testing
Sparc32	Testing
ARM	Testing
S390	Testing
Sparc64	Dev only
ia64	Dev only
m68k	Dev only

Segun la pagina web del proyecto, emula unas 64 veces mas rápido que el Bochs. Va igual de rápido que el Vmware, pero con la ventaja de que es libre y que permite hacer mas cosas ya que el soporte de arquitecturas virtuales es mayor.

Además es como una MV y un emulador al mismo tiempo ya que no solo sirve para emular un sistema completo, sino que permite ejecutar programas de una arquitectura (por ejemplo i386), en otra distinta (por ejemplo ppc).

Tipo de Licencia: GNU Library o Lesser General Public License(LGPL)
Estado de Desarrollo: 2 (Desarrollo)
Última Versión: 0.6.1 – (Nov 14, 2004)
Pagina Oficial: <http://fabrice.bellard.free.fr/qemu/>

Java Virtual Machine

La máquina virtual de Java es un programa ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico.

Empero, los intentos de la compañía propietaria de Java y productos derivados, que es Sun Microsystems, de construir microprocesadores que aceptaran el Java bytecode como su lenguaje de máquina fueron más bien infructuosos.

Actualmente la mayoría de las plataformas de ejecución de Java son máquinas virtuales. La máquina virtual es un programa normal solamente ejecutable en un sistema operativo particular (por ejemplo, Windows), sin embargo, la sola disponibilidad de la máquina virtual en nuestra plataforma de interés, permite a cualquier programa Java de ejecutarse en nuestra plataforma.

Existen varias versiones, en orden cronológico, de la máquina virtual de Java. En general la definición del Java bytecode no cambia significativamente entre versiones, y si lo hacen, los desarrolladores del lenguaje procuran que exista compatibilidad hacia atrás con los productos anteriores.

Programas de Emuladores

Algunos de los emuladores mas difundidas en sistemas Linux son:

- **Wine**
- **CrossOver Office**
- **Emuladores de Consolas de Videojuegos y Arcades**
- **Emuladores de PCs obsoletos**

WINE

WINE es un proyecto para permitir ejecutar programas diseñados para la API de Microsoft Windows bajo sistemas operativos de la familia Unix.

El nombre deriva del acrónimo recursivo "Wine No es un Emulador". Técnicamente, la afirmación del nombre es cierta, pero para muchos de sus usuarios WINE se comporta de manera similar a un emulador, lo que produce una expansión alternativa del acrónimo como "WINDows Emulator" (emulador de Windows).

WINE es software libre bajo los términos de la licencia LGPL.

La razón por la cual WINE no es un emulador es que los emuladores tienden a duplicar el entorno completo en el que un programa vive, incluyendo la simulación de una arquitectura de procesador determinada. WINE, por el contrario, implementa lo que podría ser llamado una capa de compatibilidad, la cual provee alternativas a las bibliotecas de Windows.

Wine provee un kit de herramientas (WineLib) para portar código Windows

a Unix y un cargador de aplicaciones que permite que varios programas de windows sin modificación de ningún tipo se puedan ejecutar en Unixes basados en arquitectura x86, como por ejemplo Linux, FreeBSD, y Solaris.

Historia de Wine

El proyecto WINE comenzó en 1993 para programas del sistema "Microsoft Windows 3.11". El proyecto posiblemente se originó en discusiones en comp.os.linux, los programadores Eric Youngdale y Bob Amstadt crearon su primera versión.

Hacia comienzos del 2003 WINE puede ejecutar muchos programas populares, como Lotus Notes y algunas versiones de Microsoft Office, con comportamientos y estabilidad variables. El éxito del funcionamiento de cada aplicación dependen del uso de bibliotecas dinámicas (DLL) de Windows. La empresa de software Corel ha ayudado mucho al proyecto, empleando temporalmente a uno de los principales desarrolladores del proyecto, Alexandre Julliard, junto con muchos otros programadores secundarios. Esta ayuda fue motivada por el porte de la suite ofimática de Corel a Linux. Sin embargo, debido a dificultades económicas el apoyo de Corel ha cesado.

Otras organizaciones han hecho esfuerzos comerciales para apoyar el proyecto, incluyendo CodeWeavers, Lindows y TranGaming.

CodeWeavers ha desarrollado una versión de WINE específicamente diseñada para ejecutar Microsoft Office y la comercializa bajo el nombre CrossOver Office (Ver apartado).

TransGaming produce una versión de WINE diseñada para ejecutar juegos de Windows.

El desarrollo oficial de WINE está orientado hacia la correcta implementación de la API de Windows como un todo, y se encuentra un poco atrasado en estas áreas.

El proyecto tiene una gran demanda de tiempo y posee grandes dificultades para los desarrolladores, o al menos en parte debido a la documentación incompleta de la API de Windows.

A pesar de estar la mayoría de las funciones API Win32 correctamente

documentadas, existen aún muchas áreas como formatos de archivos y protocolos para los cuales no existen especificaciones de Microsoft.

Tipo de Licencia: GNU Library o Lesser General Public License(LGPL)
Ultima Versión: 01 de diciembre del 2004
Interfase de usuario: X11
Pagina Oficial: www.winehq.com

CrossOver Office

CrossOver Office esta basado en Wine y su objetivo es poder correr aplicaciones ofimáticas y de diseño gráfico de S.O. Microsoft en sistemas Linux.

Ha logrado hacer correr en Linux entre otras, aplicaciones como:

- Microsoft Office
- Lotus Notes
- Microsoft Project
- Visio
- Macromedia Dreamweaver MX
- Flash MX
- Adobe Photoshop

Además permite usar varios plugIns para navegadores Windows en navegadores Linux tales como QuickTime y Shockwave

El uso de CrossOver Office no requiere licencias de uso de sistemas operativos Windows ya que CrossOver es un completo remplazo del SO Windows en lo que se refiere a las aplicaciones para Windows que corren en Linux

Existen dos versiones de CrossOver Office: Professional y Standard.

La versión estandard esta pensada para usuarios particulares mientras que la otra para usuarios empresariales. Se diferencian en el soporte que posee cada una de ellas y en que la versión Profesional puede ejecutar CrossOver en modo compartido desde una sola maquina.

CrossOver Office esta pensada como una herramienta de migración de Windows a Linux.

Tipo de Licencia:	Comercial
	U\$\$ 39.95 version estandard (download)
	U\$\$ 74.95 version profesional (download)
	U\$\$ 84.95 version profesional (CD)
Interfase de usuario:	X11
Pagina Oficial:	http://www.codeweavers.com/

Emuladores de Consolas de Videojuegos y Arcades

Como bien comentamos al principio de este documento hay un sinfín de emuladores de consolas de juegos y arcades. Si bien entendemos que no es el objetivo de este trabajo el analizar cada uno de ellos les comentaremos los emuladores que hemos probado y consideramos mejores o son elegidas por los usuarios Linux como las más populares.

Debemos destacar que todos ellos poseen licencia LGPL

Consolas de VideoJuegos:

Consolas Nintendo

VisualBoyAdvance es el emulador más popular para las consolas portátiles GameBoy y GameBoy Advance

Pagina de Proyecto: <http://sourceforge.net/projects/vba/>

ZSNES es el emulador más popular para las consolas SuperNintendo.

Pagina de Proyecto: <http://sourceforge.net/projects/zsnes/>

1964 es el emulador más popular para las consolas Nintendo64

Pagina de Proyecto: <http://sourceforge.net/projects/schibo/>

Nestopia es el emulador más popular para las consolas Nintendo Family Game

Pagina de Proyecto: <http://sourceforge.net/projects/nestopia/>

Consolas Atari

Stella es un emulador multiplataforma para las viejas consolas de 8k Atari 2600

Página de Proyecto: <http://sourceforge.net/projects/stella/>

Consolas SONY

P.E.Op.S. es un emulador para plataformas x86 de la Play Station One

Página de Proyecto: <http://sourceforge.net/projects/peops/>

PCSX2: emulador para Play Station 2

Página de Proyecto: <http://www.pcsx2.net>

openMSX es el emulador más popular para las viejas computadoras MSX de Sony

Página de Proyecto: <http://sourceforge.net/projects/openmsx/>

Consolas SEGA

Yabause es un emulador de Sega Saturn.

Página del Proyecto: <http://sourceforge.net/projects/yabause/>

Gens es el emulador para las consolas Sega Genesis, Sega CD y Sega 32X

Página del Proyecto: <http://sourceforge.net/projects/gens/>

Emuladores de Arcade

Los emuladores preferidos de estos juegos son dos:

XMame (<http://x.mame.net/>) es un emulador capaz de correr más de 2500 juegos muy bien documentado, actualizado y con una gran cantidad de frontends y utilidades.

Las plataformas para las cuales está soportado son las siguientes:

- Linux

- FreeBSD
- NetBSD
- Solaris (SunOS)
- OpenStep (noasm only on i386)
- IRIX
- AIX (sound currently broken)
- generic unix, no sound



AdvanceMAME / AdvanceMenu / AdvanceCD es un proyecto que desarrollo un LiveCD booteable en Linux que es una lanzador de juegos Arcade. Posee una excelente resolución de video y muy buen sonido además de un menú gráfico con iconos de los juegos que se pueden jugar. Posee además scripts para GNU/Linux, Mac OS X, DOS, Windows que permiten crear el Live CD agregandole los juegos que queramos.

Es realmente bueno ya que te permite tener en un CD booteable los juegos que quieras.

También se puede instalar solamente el puerto para el emulador MAME y MESS con o sin el Menú gráfico para correrlo en Linux sin necesidad de bootear con el LiveCD.

La página del proyecto es: <http://advancemame.sourceforge.net/>

Emuladores de PCs obsoletos

En este punto, queremos comentar que existe un sinfín de emuladores de las viejas PC hogareñas de los años 80 y 90.

Computadoras Atari

Software	Pagina del Proyecto	Arquitectura Emulada
Atari800	http://atari800.atari.org/	Atari 800
atarux	http://atarux.sourceforge.net/	Atari 800
ACE (Atari Comp. Emulator)	http://www.csh.rit.edu/[..]software.html	Atari 800
Steem	http://steem.atari.st/	Atari ST
sc68	http://sc68.atari.org/index.html	Atari ST
Hatari	http://hatari.sf.net/	Atari ST
STonX	http://www.complang.tuwien.ac.at/[..]no/stonx.html	Atari ST

Computadoras Amstrad

Software	Pagina del Proyecto	Arquitectura Emulada
CPC4X	http://ulrich-cordes.gmxhome.de/[..]sh/cpcemu.htm	Amstrad CPC
Caprice32	http://www.caprice32.cybercube.com/	Amstrad CPC
Joyce and Anne	http://www.seasip.demon.co.uk/Unix/Joyce/	Amstrad PCW

Commodore

Software	Pagina del Proyecto	Arquitectura Emulada
VICE	http://www.viceteam.org/	Commodore 8-bit
Frodo - C64 emulator	http://www.uni-mainz.de/[...]2/FRMain.html	Commodore 64
ALEC64	http://www.ibiblio.org/[...]ec/index.html	Commodore 64
Come Back 64/UNIX	http://www.classicgaming.com/cb64/cb64unix.shtml	Commodore 64
ec64	http://mars.wiwi.uni-halle.de/ec64/	Commodore 64
MyMiggy	http://mymiggy.sourceforge.net/	Amiga
The UAE Amiga Emulator	http://www.freiburg.linux.de/%7Euae/	Amiga

Sinclair ZX Spectrum

Software	Pagina del Proyecto	Arquitectura Emulada
Spectemu	http://www.inf.bme.hu/[...]jedi/spectemu/	ZX Spectrum
FBZX	http://www.rastersoft.com/[...]fbzxesp.html	ZX Spectrum
z81	http://rus.members.beeb.net/z81.html	ZX Spectrum
Glukalka	http://glukalka.sourceforge.net/	ZX Spectrum
xz80	http://users.comlab.ox.ac.uk/[...]jier/Spectrum/	ZX Spectrum
FUSE	http://fuse-emulator.sourceforge.net/	ZX Spectrum
Z80Em	http://www.komkon.org/[...]jel/misc.html	ZX Spectrum
ZxLink	http://www.zx-spectrum.net/zxlink/	ZX Spectrum
XZX-Pro	http://www.zx-spectrum.net/xzx/	ZX Spectrum

Conclusiones

Al momento de realizar una recomendación sobre los productos analizados tuvimos en cuenta los siguientes factores:

- Performance
- Confiabilidad
- Grado de madurez del software (desarrollo)
- Uso que se le va a dar al Software
- De acuerdo a lo anterior, soporte
- Tipo de Licencia

Entendimos que para el caso de maquinas virtuales en el cual se quiere tener en un entorno Linux corriendo bajo una ventana uno o más sistemas operativos hay hoy por hoy 2 posibilidades:

Posibilidad 1: Emulación de SO de Arquitectura x86:

Para esta posibilidad lo mas adecuado es el empleo de vmware, ya que es un software que resuelve todas las necesidades de maquinas virtuales de arquitectura x86 con un alto grado de performance y confiabilidad.

El unico contra que tiene es licencia ya que no es LGPL, sino comercial.

Pero por el contrario brinda un alto grado de soporte, soluciones corporativas de administración de sistemas virtuales muy interesantes y su precio para uso comercial es accesible (menos de U\$S 200 por c/ licencia workstation).

Sin embargo, si son de la idea de tener solamente software libre y necesitan maquinas virtuales de SO Windows tienen la posibilidad de utilizar Bochs en conjuncion con Plex86 para mejorar su performance. Con esta solución no se obtiene la mejor performance ni tan alta confiabilidad puesto que estas MV no estan en un estado de desarrollo estable.

Posibilidad 2: Emulación de SO de Arquitectura Libre:

Si lo que queremos es una Maquina virtual que nos permita emular arquitecturas

de procesadores tales como: x86, Sparc, ARM y PowerPC la solución es sin duda el software francés QEmu.

No solo es libre y gratuito sino que además es bastante rapido. Su unica contra radica en su madurez ya que el estado de su desarrollo no es final.

En lo que respecta a emuladores para correr determinado software, la solución es más clara, y esta dada por el Sistema Operativo y arquitectura origen de lo que queramos ejecutar en Linux.

Si se trata de aplicaciones de Windows tales como Lotus Notes u Ofimáticas, la respuesta es Wine, ya que se trata de un software maduro, libre y gratuito que logra emular una amplia gama de productos con un buen grado de estabilidad.

CrossOver Office es otra posibilidad interesante, por eso la incluimos en nuestro análisis, y puede ser válida como solución si bien es de licencia comercial para el caso que se requiera soporte en la implantación de la solución de emulación.