

Sistemas Operativos

Virtualización

Índice

Pág.

3.1. Definición y beneficios de Virtualización	3
3.1.1. Beneficios	4
3.2. Herramientas de virtualización	5
3.3. Modelos de virtualización	8
3.3.1. Bare Metal	8
3.3.2. Alojados	9
3.3.3. Modelo de paso a traves	11
3.3.4. Modelo directo de hipervisor	12
3.3.5. Modelo de controlador Dividido	13
3.4. Emulación, Simulación. Virtualización y Paravirtualización	15
3.5. Tipos de Virtualización	16
3.5.1. Virtualización de escritorio	17
3.5.2. Virtualización de redes	17
3.5.3. Virtualización de almacenamiento	18
3.5.4. Virtualización de datos	18
3.5.5. Virtualización de aplicaciones	19
3.5.6. Virtualización de centro de datos	19
3.5.7. Virtualización de CPU	20
3.5.8. Virtualización de GPU	20
3.5.9. Virtualización de Linux	21
3.5.10. Virtualización en la nube	21
3.6. Comandos linux	22
Recursos complementarios	35
Bibliografía	36

Para ejecutar simultáneamente diversas cargas de trabajo en el mismo hardware físico sin requerir que se use un solo sistema operativo para todas las aplicaciones, fue necesario crear un modelo en el que se puedan implementar simultáneamente múltiples sistemas operativos en la misma plataforma de hardware. La solución es la virtualización.

Con la virtualización, cada uno de los sistemas operativos que se implementan en la misma plataforma física tiene la ilusión de que realmente posee los recursos de hardware subyacentes, en su totalidad o en parte.

Cada sistema operativo, más sus aplicaciones y sus recursos virtuales, se denomina máquina virtual (VM). En ocasiones, las máquinas virtuales se denominan invitados o dominios .

Soportar la coexistencia de múltiples VM en una sola máquina física requiere una funcionalidad subyacente para lidiar con la asignación y administración de los recursos de hardware reales. Además, es necesario proporcionar algunas garantías de aislamiento en las máquinas virtuales.

Esta funcionalidad la proporciona la capa de virtualización, también denominada monitor de máquina virtual o hipervisor .

3.1. Definición y beneficios de Virtualización

Una máquina virtual es un **duplicado eficiente y aislado** de una máquina real.

La virtualización es compatible con el monitor de máquina virtual (VMM). El VMM tiene tres responsabilidades.

Primero, el VMM debe proporcionar un entorno que sea esencialmente idéntico al de la máquina original. La capacidad puede diferir, pero la configuración general (tipo de CPU, tipos de dispositivos) debe ser la misma. El VMM debe proporcionar cierta **fidelidad** a que la representación del hardware

que es visible para la VM coincida con el hardware que está disponible en la plataforma física.

En segundo lugar, los programas que se ejecutan en máquinas virtuales deben mostrar, en el peor de los casos, solo pequeñas disminuciones de velocidad. Claramente, las VM solo reciben una parte de los recursos disponibles para la máquina host. Sin embargo, el objetivo de VMM es garantizar que la VM funcione a la misma velocidad que una aplicación nativa si se le dieran todos los recursos del host. El VMM debe proporcionar un **rendimiento** a las máquinas virtuales lo más cercano posible al rendimiento nativo.

Finalmente, el VMM tiene el control total de los recursos del sistema. El VMM controla quién accede a qué recursos y cuándo, y se puede confiar en él para brindar **seguridad y aislamiento** entre las VM.

3.1.1. Beneficios:

La virtualización permite la **consolidación**. La consolidación se refiere a la capacidad de ejecutar varias máquinas virtuales en una única plataforma física. La consolidación conduce a menos máquinas, con menos espacio, con menos administradores, con facturas de electricidad potencialmente más pequeñas que aún pueden ejecutar el mismo grado / tipo de carga de trabajo.

La consolidación nos permite disminuir costos y mejorar la manejabilidad.

La virtualización también facilita la **migración**. Dado que el sistema operativo y las aplicaciones ya no están acoplados al sistema físico, la virtualización facilita la configuración, el desmontaje y la clonación de máquinas virtuales.

La virtualización nos ayuda a abordar la **disponibilidad** y la **confiabilidad**. Si notamos que una máquina física está a punto de dejar de funcionar, podemos

poner en marcha fácilmente nuevas máquinas virtuales en una plataforma física diferente.

Debido a que el sistema operativo y las aplicaciones están bien encapsulados en una máquina virtual, es más fácil contener errores o códigos maliciosos en esos contenedores aislados sin derribar otras máquinas virtuales o toda la plataforma física.

La virtualización también ha proporcionado un terreno fértil para la investigación de sistemas operativos. La investigación puede arrancar rápidamente y probar los sistemas operativos en desarrollo sin tener que reiniciar el hardware.

La virtualización también proporciona un soporte asequible para los sistemas operativos heredados. Las aplicaciones que deben ejecutarse en sistemas operativos más antiguos ya no necesitan una caja física dedicada solo a ellas. Pueden ejecutarse como una VM entre muchas, compartiendo el recurso físico con otras aplicaciones.

3.2. Herramientas de virtualización

Las herramientas de virtualización han sido inmensamente el tema de investigación ya que ayudan a lograr la virtualización en muchos tipos de hardware. Las herramientas de virtualización más utilizadas están relacionadas principalmente con los productos de VMware, que son VMware Vcenter Converter y VMware Workstation. Algunas de las herramientas son de código abierto, es decir, están disponibles de forma gratuita para cualquiera, excepto el VMware Server, que es gratuito pero no de código abierto (el código no es accesible para todos), debido a que tiene dos plataformas de hardware, es decir, Linux y Windows, mientras que las herramientas de virtualización, como Microsoft Virtual PC o cualquier otra herramienta, están limitadas a un solo sistema operativo o plataforma de hardware, lo que significa que están

restringidas a sus propios tipos de software. Es importante tener en cuenta que hay algunas similitudes entre el entorno real y el entorno virtual que dependen indirectamente de la técnica utilizada para la virtualización. Algunas de las herramientas de virtualización más populares utilizadas en varios campos de la informática son las siguientes:

1. VNUML es un código abierto y está disponible para todos los usuarios para su descarga gratuita. VNUML es básicamente una herramienta de virtualización utilizada para múltiples sistemas virtuales del sistema operativo Linux. Estos sistemas virtuales se conocen como invitados que ejecutan sus aplicaciones junto con el sistema operativo Linux del sistema original que se denomina host.

2. Virtual Box se utiliza para la implementación de máquinas virtuales en los ordenadores y servidores físicos. También realiza una virtualización completa en el ordenador anfitrión, lo que significa que sin ninguna modificación en el sistema operativo, el sistema operativo invitado se ejecuta en el ordenador anfitrión.

3. VMware Server es una herramienta de virtualización de código libre para el sistema operativo Linux y Windows. VMware Server se basa en la virtualización completa, es decir, el ordenador de sobremesa físico para ejecutar más de una máquina virtual de diferentes sistemas operativos llamados invitados en él.

4. Qemu: Qemu se utiliza para la ejecución de la virtualización en los sistemas operativos como Linux y Windows. Es un popular emulador de código abierto que proporciona una rápida emulación con la ayuda de la traducción dinámica. Tiene muchos comandos útiles para la gestión de VM.

5. Xen: es también una herramienta de código abierto para la virtualización utilizada ampliamente para la virtualización de Para en el PC anfitrión y los ordenadores invitados.

6. VMware es una plataforma VM (máquina virtual) que ayuda a la ejecución del sistema operativo sin modificar en el hardware de la aplicación de nivel de host o de usuario. El sistema operativo que se ejecuta con VMware puede bloquearse, reinstalarse, reiniciarse o bloquearse sin que ello afecte a la aplicación que se ejecuta en el ordenador anfitrión. VMware permite separar el sistema operativo invitado del sistema operativo anfitrión real, de modo que si el sistema operativo invitado falla, el hardware físico o la máquina anfitriona no sufre las consecuencias. Por lo tanto, el VMware se utiliza para ejecutar varios sistemas operativos no modificados al mismo tiempo en la única máquina de hardware mediante la ejecución del sistema operativo en la máquina virtual del sistema operativo específico. En lugar de la ejecución indirecta de código en el hardware como en el caso del simulador de software, la máquina virtual ejecuta el código directamente en el hardware físico sin ninguna aplicación para la interpretación del código.

7. La herramienta de virtualización **EMF** es un plug in basado en eclipse para mantener el uso transparente de los modelos virtuales, todos ellos basados en EMF. Para la creación de un modelo virtual utilizando la herramienta EMF, los usuarios tienen que proporcionar los modelos que contribuyen junto con los Meta modelos para la virtualización. Los siguientes tres elementos son los básicos de cualquier modelo virtual formado por la herramienta EMF.

- El usuario puede definirlo o puede ser la amalgama de varios procesos de composición separados.
- Modelo de correspondencia se define principalmente junto con la herramienta AMW2. Este modelo de correspondencia contiene todos los enlaces virtuales que se relacionan en los elementos que contribuyen e identifican de qué manera se van a componer.
- Modelo virtual es un archivo que especifica la ubicación física de todos los recursos de hardware que se van a utilizar en el proceso de composición virtual.

8. Virtual EMF, la especificación de esta herramienta es que permite superar las limitaciones de los modelos virtuales, como que los modelos virtuales no pueden soportar datos concretos aunque sean fácilmente accesibles. Ayudan a manipular los datos originales contenidos en otros modelos de EMF, por lo que esta herramienta también está construida sobre Eclipse/EMF1.

3.3. Modelos de virtualización

3.3.1. Bare metal

En la virtualización **bare-metal** (también conocida como virtualización **basada en hipervisor** o de **tipo 1** , VMM administra todos los recursos de hardware y admite la ejecución de las máquinas virtuales.

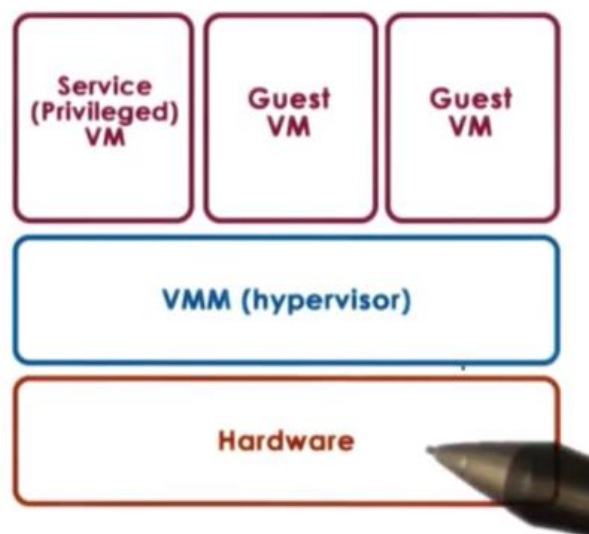


Figura 1. Modelo de virtualización bare metal.

Un problema con este modelo se refiere a los dispositivos. Según el modelo, el hipervisor debe gestionar todos los dispositivos posibles. En otras palabras, los fabricantes de dispositivos ahora tienen que proporcionar controladores de dispositivos no solo para diferentes sistemas operativos, sino también para diferentes hipervisores.

Para eliminar este problema, el modelo de hipervisor típico integra una máquina virtual especial, una VM de servicio, que ejecuta un sistema operativo estandarizado con privilegios completos de acceso al hardware, lo que le permite manipular el hardware como si fuera nativo.

La VM privilegiada ejecuta todos los controladores de dispositivos y controla cómo se utilizan los dispositivos en la plataforma. Esta máquina virtual ejecutará algunas otras tareas de configuración y administración para ayudar aún más al hipervisor.

Este modelo está adaptado por el software de virtualización **Xen** y por el hipervisor **ESX** de VMware.

En cuanto a Xen, las máquinas virtuales que se ejecutan en el entorno virtualizado se denominan dominios. El dominio privilegiado se denomina **dom0** y los dominios invitados se denominan **domU**. Xen es el hipervisor. Todos los controladores de dispositivos se ejecutan en dom0.

VMware mantiene la mayor proporción de núcleos de servidor. VMware ha podido exigir que los fabricantes de dispositivos desarrollen controladores que puedan ser utilizados por el hipervisor. Dado que VMware se dirige a los servidores, hay relativamente pocos dispositivos. Para apoyar a la comunidad de código abierto, VMware tiene muchas API de código abierto.

VMware solía tener un núcleo de control basado en Linux (similar al dom0 en Xen), pero ahora toda la configuración se realiza a través de API remotas.

3.3.2. Alojados

El otro tipo de modelo de virtualización es el modelo **alojado** (o **tipo 2**).

En este modelo, hay un sistema operativo host completo que administra todos los recursos de hardware. El sistema operativo host integra un VMM, que

es responsable de proporcionar a las máquinas virtuales su interfaz de plataforma virtual.

Según sea necesario, el módulo VMM invocará los controladores de dispositivos y otros componentes del host según sea necesario.

Un beneficio de este modelo es que puede aprovechar todos los servicios y mecanismos que ya están desarrollados para el sistema operativo host. Se necesita desarrollar mucha menos funcionalidad para el módulo VMM en sí.

En esta configuración, puede ejecutar máquinas virtuales invitadas a través del módulo VMM, así como aplicaciones nativas directamente en el sistema operativo host.

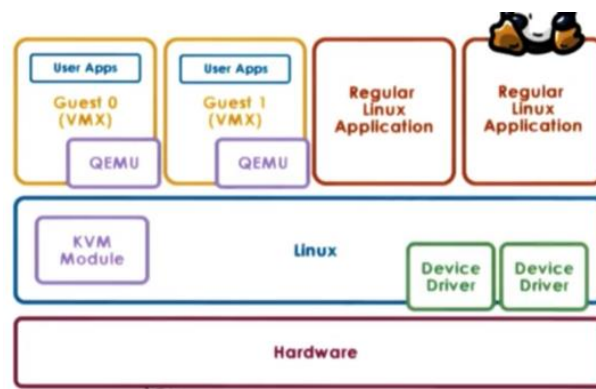


Figura 2. Modelo de Virtualización Alojada

Un ejemplo del modelo alojado es la **máquina virtual basada en kernel (KVM)** que está integrada en Linux. El host Linux proporciona todos los aspectos de la gestión del hardware físico y puede ejecutar aplicaciones Linux normales directamente.

El soporte para ejecutar VM invitadas proviene de una combinación del módulo KVM (VMM) y un emulador de hardware llamado **QEMU**.

QEMU se utiliza como una interfaz virtual entre la VM y el hardware físico, y solo interviene durante cierto tipo de instrucciones críticas; por ejemplo, gestión de E / S.

KVM ha podido aprovechar todos los avances que se han contribuido a la comunidad de código abierto de Linux. Debido a esto, KVM puede adaptarse rápidamente a nuevas funciones y correcciones.

3.3.3. Modelo de paso a través

En el **modelo de paso a través**, el controlador de nivel VMM es responsable de configurar los permisos de acceso a un dispositivo. Por ejemplo, permitirá que una máquina virtual invitada tenga acceso a los registros de memoria correspondientes a un dispositivo.

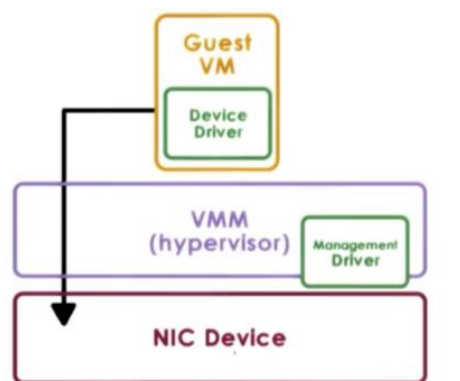


Figura 3. Modelo de de paso a traves

En este enfoque, la máquina virtual invitada tiene acceso exclusivo a un dispositivo. Además, la VM puede acceder directamente al dispositivo, sin interactuar con el VMM. Este modelo también se denomina **modelo de derivación VMM**.

El problema con este acceso exclusivo es que dificulta compartir dispositivos entre máquinas virtuales. El hipervisor deberá reasignar continuamente a quién pertenece el dispositivo a lo largo del tiempo, y el

acceso al dispositivo no ocurrirá simultáneamente en las VM. Esto a menudo es inviable.

Debido a que el hipervisor está completamente fuera del camino, la VM invitada y el controlador de dispositivo en la VM invitada operan y controlan directamente el dispositivo. Esto significa que debe haber un dispositivo *exactamente del mismo tipo* en la plataforma física que espera el sistema operativo invitado.

Uno de los beneficios de la virtualización es que las VM invitadas están desacopladas del hardware físico, lo que facilita la migración de invitados a través de nodos físicos.

Este modelo de paso a través rompe ese desacoplamiento al vincular un dispositivo a una VM. Esto reintroduce la complejidad de la migración, ya que puede haber algún estado residente / específico del dispositivo que deba copiarse y configurarse correctamente en un modo de destino.

3.3.4. Modelo directo de hipervisor

En el **modelo directo de hipervisor**, el hipervisor intercepta todas las solicitudes de acceso al dispositivo que realiza la máquina virtual invitada.

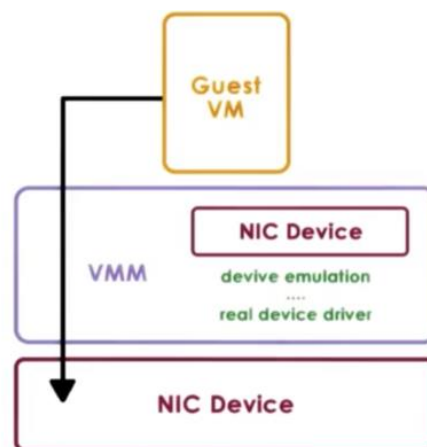


Figura 4. Modelo de directo de hypervisor

El hipervisor ya no tiene la restricción de que el dispositivo solicitado y el dispositivo físico coincidan.

En cambio, el hipervisor puede traducir la solicitud de acceso al dispositivo en alguna representación genérica de una operación de E / S para esa familia de dispositivos en particular (red o disco, por ejemplo) y luego atravesar la pila de E / S residente del hipervisor. La parte inferior de esa pila es el controlador de dispositivo real real, que el hipervisor finalmente invocará para cumplir con la solicitud.

Un beneficio clave de este enfoque es que la VM permanece desacoplada de la plataforma / dispositivo físico. Como resultado, la migración sigue siendo fácil. Además, ahora se pueden compartir dispositivos, ya que el hipervisor administra el acceso a los dispositivos.

La desventaja del modelo es que el paso de emulación del dispositivo agrega latencia a los accesos al dispositivo. Además, este modelo requiere que el hipervisor admita todos los controladores para que pueda realizar las emulaciones necesarias, lo que significa que el hipervisor ahora está expuesto a todas las complejidades y complicaciones de varios controladores de dispositivos.

3.3.5. Modelo de controlador de dispositivo dividido

En el **modelo de controlador de dispositivo dividido** , todos los accesos a los dispositivos se controlan de una manera que involucra un componente que reside en una VM invitada y un componente que reside en una capa de hipervisor.

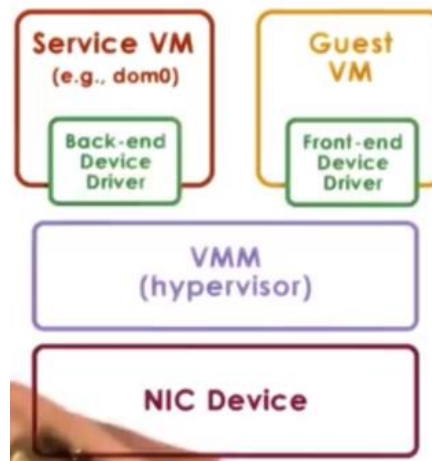


Figura 5. Modelo de controlador de dispositivo dividido

El **controlador de front-end** reside en la VM invitada y el controlador real del dispositivo físico, el **controlador de back-end**, reside en la VM de servicio (o el host en la virtualización de tipo 2).

Aunque el controlador de back-end no tiene que modificarse necesariamente, ya que es el mismo controlador que usaría el sistema operativo como si se ejecutara de forma nativa, el controlador de front-end sí debe modificarse.

El controlador de front-end debe tomar las operaciones del dispositivo que realizan las aplicaciones en el invitado y luego envolverlas en un formato estándar para entregarlas al componente de back-end. Debido a esta modificación, este modelo solo se puede utilizar realmente en huéspedes paravirtualizados.

Una ventaja de este enfoque es que se puede eliminar la sobrecarga de emulación de dispositivos. Otro beneficio de este enfoque es que el componente de back-end centralizado permite una mejor gestión de los dispositivos compartidos

3.4. Emulación, Simulación. Virtualización y Paravirtualización

La virtualización es tomar un sistema operativo y colocar un intermediario (conocido como hipervisor) entre él y el hardware. Lo que significa que puede ejecutar varios sistemas operativos simultáneamente en un conjunto de hardware.

Un emulador permite que un tipo de sistema operativo ejecute aplicaciones para diferentes sistemas operativos. Entonces, por ejemplo, existen emuladores para Commodore Amiga o Atari ST para PC.

Una simulación, en este contexto, es un software escrito para que se vea y actúe como otro fragmento de software, tal vez simulando estar en un hardware diferente. Un ejemplo de esto son los simuladores de iOS para equipos de red de Cisco, que le permiten practicar los comandos sin comprar un conmutador / enrutador.

Paravirtualización funciona de manera diferente a la virtualización completa. No necesita simular el hardware de las máquinas virtuales. El hipervisor está instalado en un servidor físico (host) y un sistema operativo invitado está instalado en el entorno. Invitados virtuales conscientes de que se ha virtualizado, a diferencia de la virtualización completa (donde el huésped no sabe que se ha virtualizado) para aprovechar las funciones. En este método de virtualización, los códigos fuente de los invitados se modificarán con información confidencial para comunicarse con el host. Los sistemas operativos invitados requieren extensiones para realizar llamadas API al hipervisor. En la virtualización completa, los invitados emitirán llamadas de hardware, pero en la paravirtualización, los invitados se comunicarán directamente con el host (hipervisor) utilizando los controladores. Aquí está la lista de productos que apoyan la paravirtualización.

- Xen
- IBM LPAR
- Oracle VM para SPARC (LDOM)
- Oracle VM para X86 (OVM)

El siguiente diagrama puede ayudarlo a comprender cómo Xen admite tanto la virtualización completa como la paravirtualización. Debido a la diferencia de arquitectura entre Windows y el hipervisor Xen basado en Linux, el sistema operativo Windows no se puede paravirtualizar. Pero lo hace para los invitados de Linux modificando el kernel. Pero VMware ESXi no modifica el kernel para los invitados de Linux y Windows.

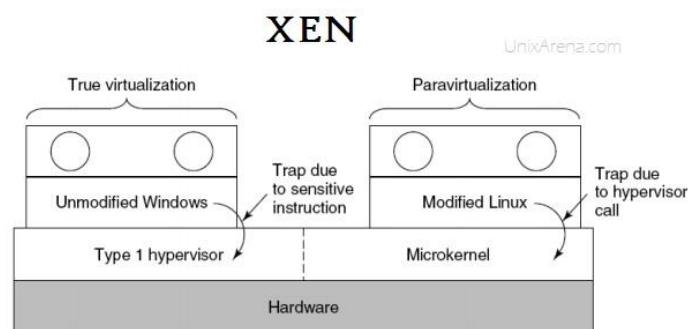


Figura 6. Xen (Virtualización y paravirtualización).

Xen admite tanto la virtualización completa como la paravirtualización

3.5. Tipos de Virtualización

Hasta este punto, hemos hablado de la virtualización de servidores, pero muchos otros elementos de la infraestructura de TI se pueden virtualizar para brindar ventajas significativas a los administradores de TI (en particular) y a la empresa en su conjunto. En esta sección, cubriremos los siguientes tipos de virtualización:

- Virtualización de escritorio
- Virtualización de redes
- Virtualización de almacenamiento

- Virtualización de datos
- Virtualización de aplicaciones
- Virtualización del centro de datos
- Virtualización de CPU
- Virtualización de GPU
- Virtualización de Linux
- Virtualización en la nube

3.5.1. Virtualización de escritorio

La virtualización de escritorio le permite ejecutar varios sistemas operativos de escritorio, cada uno en su propia máquina virtual en la misma computadora.

Hay dos tipos de virtualización de escritorio:

- **La infraestructura de escritorio virtual (VDI)** ejecuta varios escritorios en máquinas virtuales en un servidor central y los transmite a los usuarios que inician sesión en dispositivos de cliente ligero. De esta manera, VDI permite que una organización proporcione a sus usuarios acceso a una variedad de sistemas operativos desde cualquier dispositivo, sin instalar sistemas operativos en ningún dispositivo. Consulte " ¿Qué es la infraestructura de escritorio virtual (VDI)? " Para obtener una explicación más detallada.
- **La virtualización de escritorio local** ejecuta un hipervisor en una computadora local, lo que permite al usuario ejecutar uno o más sistemas operativos adicionales en esa computadora y cambiar de un sistema operativo a otro según sea necesario sin cambiar nada sobre el sistema operativo principal.

3.5.2. Virtualización de redes

La virtualización de red utiliza software para crear una "vista" de la red que un administrador puede usar para administrar la red desde una única consola. Extrae elementos y funciones de hardware (por ejemplo, conexiones, conmutadores, enrutadores, etc.) y los abstrae en software que se ejecuta en un hipervisor. El administrador de la red puede modificar y controlar estos elementos sin tocar los componentes físicos subyacentes, lo que simplifica drásticamente la gestión de la red.

Los tipos de virtualización de red incluyen **redes definidas por software (SDN)** , que virtualizan el hardware que controla el enrutamiento del tráfico de la red (llamado "plano de control"), y **la virtualización de funciones de red (NFV)** , que virtualiza uno o más dispositivos de hardware que proporcionan una red específica. función (por ejemplo, un firewall, equilibrador de carga o analizador de tráfico), lo que hace que esos dispositivos sean más fáciles de configurar, aprovisionar y administrar.

3.5.3. Virtualización de almacenamiento

La virtualización del almacenamiento permite acceder a todos los dispositivos de almacenamiento de la red , ya sea que estén instalados en servidores individuales o unidades de almacenamiento independientes, como un único dispositivo de almacenamiento y administrarlos. Específicamente, la virtualización del almacenamiento agrupa todos los bloques de almacenamiento en un solo grupo compartido desde el cual se pueden asignar a cualquier VM de la red según sea necesario. La virtualización del almacenamiento facilita el aprovisionamiento de almacenamiento para las máquinas virtuales y aprovecha al máximo todo el almacenamiento disponible en la red.

3.5.4. Virtualización de datos

Las empresas modernas almacenan datos de múltiples aplicaciones, utilizando múltiples formatos de archivo, en múltiples ubicaciones, que van

desde la nube hasta sistemas de hardware y software locales. La virtualización de datos permite que cualquier aplicación acceda a todos esos datos, independientemente de la fuente, el formato o la ubicación.

Las herramientas de virtualización de datos crean una capa de software entre las aplicaciones que acceden a los datos y los sistemas que los almacenan. La capa traduce la solicitud o consulta de datos de una aplicación según sea necesario y devuelve resultados que pueden abarcar varios sistemas. La virtualización de datos puede ayudar a romper los silos de datos cuando otros tipos de integración no son factibles, deseables o asequibles.

3.5.5. Virtualización de aplicaciones

La virtualización de aplicaciones ejecuta el software de la aplicación sin instalarlo directamente en el sistema operativo del usuario. Esto difiere de la virtualización de escritorio completa (mencionada anteriormente) porque solo la aplicación se ejecuta en un entorno virtual; el sistema operativo en el dispositivo del usuario final se ejecuta como de costumbre. Hay tres tipos de virtualización de aplicaciones:

- **Virtualización de aplicaciones locales:** toda la aplicación se ejecuta en el dispositivo de punto final, pero se ejecuta en un entorno de tiempo de ejecución en lugar de en el hardware nativo.
- **Transmisión de aplicaciones:** la aplicación reside en un servidor que envía pequeños componentes del software para que se ejecuten en el dispositivo del usuario final cuando sea necesario.
- **Virtualización de aplicaciones basada en servidor** La aplicación se ejecuta completamente en un servidor que envía solo su interfaz de usuario al dispositivo cliente.

3.5.6. Virtualización del centro de datos

La virtualización del centro de datos abstrae la mayor parte del hardware de un centro de datos en software, lo que permite a un administrador dividir un solo centro de datos físico en varios centros de datos virtuales para diferentes clientes.

Cada cliente puede acceder a su propia infraestructura como servicio (IaaS), que se ejecutaría en el mismo hardware físico subyacente. Los centros de datos virtuales ofrecen un acceso fácil a la computación basada en la nube, lo que permite que una empresa configure rápidamente un entorno de centro de datos completo sin comprar hardware de infraestructura.

3.5.7. Virtualización de CPU

La virtualización de la CPU (unidad central de procesamiento) es la tecnología fundamental que hace posibles los hipervisores, las máquinas virtuales y los sistemas operativos. Permite que una sola CPU se divida en varias CPU virtuales para que la utilicen varias máquinas virtuales.

Al principio, la virtualización de la CPU estaba completamente definida por software, pero muchos de los procesadores actuales incluyen conjuntos de instrucciones ampliados que admiten la virtualización de la CPU, lo que mejora el rendimiento de la máquina virtual.

3.5.8. Virtualización de GPU

Una GPU (unidad de procesamiento gráfico) es un procesador multinúcleo especial que mejora el rendimiento informático general al hacerse cargo del procesamiento gráfico o matemático de alta resistencia. La virtualización de GPU permite que varias máquinas virtuales usen todo o parte de la potencia de procesamiento de una sola GPU para video más rápido, inteligencia artificial (AI) y otras aplicaciones con uso intensivo de gráficos o matemáticas.

- **Las GPU de paso** hacen que toda la GPU esté disponible para un solo sistema operativo invitado.
- **Shared vGPU** s dividir núcleos GPU físicas entre varias GPU virtuales (vGPUs) para su uso por las máquinas virtuales basadas en servidor.

3.5.9. Virtualización de Linux

Linux incluye su propio hipervisor, llamado máquina virtual basada en kernel (KVM), que admite las extensiones de procesador de virtualización de Intel y AMD para que pueda crear máquinas virtuales basadas en x86 desde un sistema operativo host Linux.

Como sistema operativo de código abierto, Linux es altamente personalizable. Puede crear máquinas virtuales que ejecuten versiones de Linux adaptadas a cargas de trabajo específicas o versiones de seguridad reforzada para aplicaciones más sensibles.

3.5.10. Virtualización en la nube

Como se señaló anteriormente, el modelo de computación en la nube depende de la virtualización. Al virtualizar servidores, almacenamiento y otros recursos físicos del centro de datos, los proveedores de computación en la nube pueden ofrecer una variedad de servicios a los clientes, incluidos los siguientes:

Infraestructura como servicio (IaaS) : servidor virtualizado, almacenamiento y recursos de red que puede configurar según sus requisitos.

Plataforma como servicio (PaaS) : herramientas de desarrollo virtualizadas, bases de datos y otros servicios basados en la nube que puede utilizar para crear sus propias aplicaciones y soluciones basadas en la nube.

Software como servicio (SaaS) : aplicaciones de software que utiliza en la nube. SaaS es el servicio basado en la nube que más se abstrae del hardware.

3.6. Comandos linux

Listado de archivos (ls)

Si desea ver la lista de archivos en su sistema UNIX o Linux, use el comando 'ls' .

Muestra los archivos / directorios en su directorio actual.

```
guru99@VirtualBox:~$ ls
Desktop  Downloads  Music      Public  Videos
Documents examples.desktop Pictures  Templates
```

Figura 7. Comando ls

Nota:

- Los directorios se indican en color azul.
- Los archivos se indican en blanco.
- Encontrará esquemas de color similares en diferentes versiones de Linux.

Supongamos que su carpeta "Música" tiene los siguientes subdirectorios y archivos.

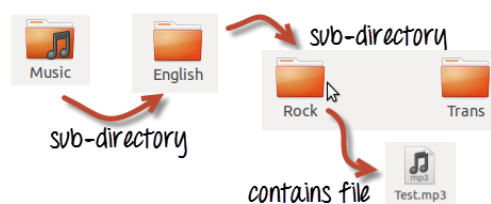


Figura 7. Ejemplo de organización de carpetas

Puede usar 'ls -R' para mostrar todos los archivos no solo en directorios sino también en subdirectorios

```
guru99@VirtualBox:~$ ls -R
.:
Desktop Downloads Music Public Videos
Documents examples.desktop Pictures Templates

./Desktop:
./Documents:
./Downloads:
./Music:
English
./Music/English:
Rock Trans
./Music/English/Rock:
Test.mp3
./Music/English/Trans:
./Pictures:
./Public:
./Templates:
./Videos:
guru99@VirtualBox:~$
```

Figura 8. Uso del comando ls -r

NOTA: El comando distingue entre mayúsculas y minúsculas. Si ingresa, "ls - r" you will get an error.

'ls -al' proporciona información detallada de los archivos. El comando proporciona información en formato de columnas. Las columnas contienen la siguiente información:

Tabla 1:

Información detallada del comando ls -al

1 ^a Columna	Tipo de archivo y permisos de acceso
2 ^a Columna	# de enlaces duros al archivo
3 ^a Columna	Propietario y creador del archivo
4 ^a Columna	Grupo del propietario
5 ^a Columna	Tamaño de archivo en bytes
6 ^a Columna	Fecha y hora
7 ^a Columna	Directorio o nombre de archivo

Veamos un ejemplo:

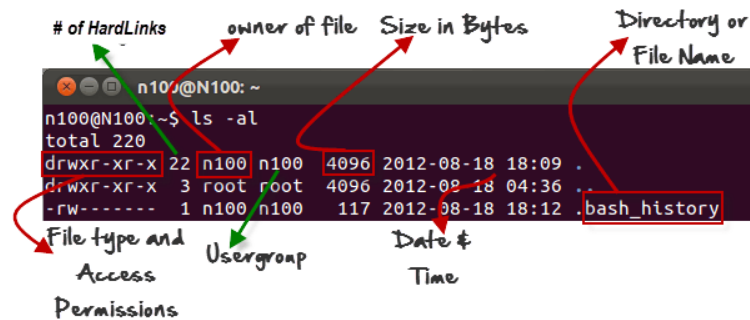


Figura 9. Uso del comando ls -al

Listar archivos ocultos

Los elementos ocultos en UNIX / Linux comienzan con '.' al comienzo del archivo o directorio.

Cualquier directorio / archivo que comience con '.' no se verá a menos que lo solicite. Para ver archivos ocultos, use el comando ls -a.

```
guru99@VirtualBox:~$ ls -a
.          .dmrc          .ICEauthority  sample
..         Documents  .local         sample1
.bash_history Downloads      .mission-control sample2
.bash_logout examples.desktop Music          Templates
.bashrc    .gconf        Pictures       .thumbnails
.cache     .gnome2       .profile       Videos
.config    .gstreamer-0.10 Public         .Xauthority
.dbus      .gtk-bookmarks .pulse         .xsession-error
Desktop    .gvfs         .pulse-cookie
guru99@VirtualBox:~$
```

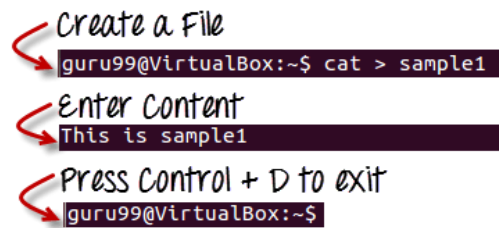
Figura 10. Uso del comando ls -a

Creación y visualización de archivos

El comando del servidor 'cat' se usa para mostrar archivos de texto. También se puede utilizar para copiar, combinar y crear nuevos archivos de texto. Vamos a ver cómo funciona.

Para crear un nuevo archivo, use el comando

- cat> nombre de archivo
- Agregar contenido
- Presione 'ctrl + d' para regresar al símbolo del sistema.



```
guru99@VirtualBox:~$ cat > sample1
This is sample1
guru99@VirtualBox:~$
```

Figura 11. Comando cat para agregar contenido

Veamos el archivo que acabamos de crear:

```
guru99@VirtualBox:~$ cat sample1
This is sample1
```

Figura 12. Visualizando un archivo con el comando cat

Veamos otro archivo sample2

```
guru99@VirtualBox:~$ cat > sample2
This is sample2
```

Figura 13. Ejemplo con el comando cat

La sintaxis para combinar 2 archivos es: `cat file1 file2 > nuevo nombre de archivo`

Combinemos la muestra 1 y la muestra 2.

```
guru99@VirtualBox:~$ cat sample1 sample2 > sample
```

Figura 14. Combinando archivos con cat

Tan pronto como inserte este comando y presione enter, los archivos se concatenan, pero no ve un resultado. Esto se debe a que Bash Shell (Terminal) es de tipo silencioso. Los comandos de Shell nunca le darán un mensaje de confirmación como "OK" o "Comando ejecutado exitosamente". Solo mostrará un mensaje cuando algo salga mal o cuando haya ocurrido un error.

Para ver el nuevo archivo combinado "sample", utilice el comando `cat sample`

```
guru99@VirtualBox:~$ cat sample
This is sample1
This is sample2
```

Figura 15. Visualización de archivos con cat

Nota: Solo se pueden mostrar y combinar archivos de texto con este comando.

Eliminar archivos

El comando 'rm' elimina archivos del sistema sin confirmación.

Para eliminar un archivo, utilice la sintaxis: `rm nombre de archivo`

The screenshot shows a terminal session with three steps, each with a handwritten red arrow pointing to the command:

- List current contents of directory:** `guru99@VirtualBox:~$ ls`. The output lists: Desktop, Downloads, Music, [Public, sample1, Templates, Documents, examples.desktop, Pictures, sample, sample2, Videos. The file 'sample1' is circled in red.
- Remove the file sample1:** `guru99@VirtualBox:~$ rm sample1`
- List directory, to check file has been deleted:** `guru99@VirtualBox:~$ ls`. The output lists: Desktop, Downloads, Music, Public, sample2, Videos, Documents, examples.desktop, Pictures, sample, Templates. The file 'sample1' is no longer present.

Figura 16. Eliminar archivos con rm

Mover y renombrar archivos

Para mover un archivo, use el comando: `mv nombre de archivo ubicación_archivo_nuevo`

Supongamos que queremos mover el archivo "sample2" a la ubicación / home / guru99 / Documents. Ejecutando el comando: `mv sample2 / home / guru99 / Documentos`

The screenshot shows a terminal command: `guru99@VirtualBox:~$ mv sample2 /home/guru99/Documents`. The output is: `mv: cannot move 'sample2' to '/home/guru99/Documents': Permission denied`. A red 'X' icon is placed to the left of the command line.

Figura 17. Aplicando el comando mv sin permiso de superusuario

El comando mv necesita permiso de superusuario. Actualmente, estamos ejecutando el comando como usuario estándar. Por lo tanto, obtenemos el error anterior. Para superar el error, use el comando: `sudo comando(cualquiera)`

El comando Sudo permite a los usuarios habituales ejecutar programas con los privilegios de seguridad del superusuario o root.

El comando Sudo solicitará la autenticación de contraseña. Sin embargo, no es necesario que conozca la contraseña de root. Puede proporcionar su propia contraseña. Después de la autenticación, el sistema invocará el comando solicitado.

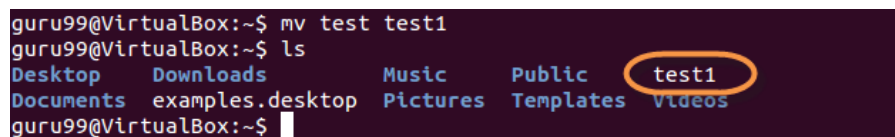
Sudo mantiene un registro de cada ejecución de comando. Los administradores del sistema pueden rastrear a la persona responsable de los cambios no deseados en el sistema.

```
guru99 @ VirtualBox: ~ $ sudo mv sample2 / home / quru99 / Documentos
```

```
[sudo] contraseña para guru99: ****
```

```
guru99 @ VirtualBox: ~ $
```

Para cambiar el nombre del archivo: `mv nombre de archivo nuevo nombre de archivo`

A terminal window with a dark purple background. The first line shows the command 'mv test test1' being executed. The second line shows the command 'ls' being executed. The output of 'ls' is a list of directories: Desktop, Downloads, Music, Public, test1, Documents, examples.desktop, Pictures, Templates, and Videos. The 'test1' directory is circled in orange. The prompt 'guru99@VirtualBox:~\$' is visible at the bottom.

```
guru99@VirtualBox:~$ mv test test1
guru99@VirtualBox:~$ ls
Desktop  Downloads  Music      Public    test1
Documents examples.desktop Pictures    Templates Videos
guru99@VirtualBox:~$
```

Figura 17. Cambiando el nombre de un archivo

NOTA : De forma predeterminada, la contraseña que ingresó para sudo se conserva durante 15 minutos por terminal. Esto elimina la necesidad de ingresar la contraseña una y otra vez.

Solo necesita privilegios de root / sudo, solo si el comando involucra archivos o directorios que no son propiedad del usuario o grupo que ejecuta los comandos.

Creando Directorios

Los directorios se pueden crear en un sistema operativo Linux usando el siguiente comando: `mkdir nombre de directorio`

Este comando creará un subdirectorio en su directorio de trabajo actual, que normalmente es su "Directorio de inicio".

Por ejemplo: `mkdir mydirectory`

```
home@VirtualBox:~$ mkdir mydirectory
home@VirtualBox:~$ ls
Desktop    Downloads  Music      Pictures   Templates
Documents  examples.desktop mydirectory Public      Videos
home@VirtualBox:~$
```

Figura 18. Comando mkdir

Si desea crear un directorio en una ubicación diferente que no sea 'Directorio de inicio', puede usar el siguiente comando: `mkdir`

Por ejemplo: `mkdir /tmp/MUSIC` creará un directorio 'Música' en el directorio '/tmp'

```
home@VirtualBox:~$ mkdir /tmp/MUSIC
home@VirtualBox:~$ ls /tmp
keyring-yCD2no  pulse-0b9vyJcXyHZz  ssh-SSSsjczv1036  virtual-home.HaC7Mw
MUSIC          pulse-PKdhtXMmr18n  unity_support_test.1
home@VirtualBox:~$
```

Figura 19. Comando mkdir y ls

También puede crear más de un directorio a la vez.

```
home@VirtualBox:~$ mkdir dir1 dir2 dir3
home@VirtualBox:~$ ls
Desktop  dir2  Documents  examples.desktop  Pictures  Templates
dir1     dir3  Downloads  Music             Public   Videos
home@VirtualBox:~$
```

Figura 19. Comando mkdir creando varios directorios

Eliminar directorios

Para eliminar un directorio, use el comando: `rmdir nombre_de_directorio`

Ejemplo: `rmdir mydirectory` eliminará el directorio mydirectory

```
home@VirtualBox:~$ rmdir mydirectory
home@VirtualBox:~$ ls
Desktop  dir2  Documents  examples.desktop  Pictures  Templates
dir1     dir3  Downloads  Music             Public   Videos
home@VirtualBox:~$
```

Figura 20. Comando rmdir

Sugerencia: asegúrese de que no haya ningún archivo / subdirectorio en el directorio que desea eliminar. Primero elimine los archivos / subdirectorio antes de eliminar el directorio principal.

```
home@VirtualBox:~$ rmdir Documents
rmdir: failed to remove 'Documents': Directory not empty
home@VirtualBox:~$
```

Figura 20. Comando rmdir con problema de eliminación

Cambio de nombre de directorio

El comando 'mv' (mover) (tratado anteriormente) también se puede usar para cambiar el nombre de los directorios. Utilice el formato que se proporciona a continuación: `mv nombre_directorio nuevo_directorio`

```
home@VirtualBox:~$ mv mydirectory newdirectory
home@VirtualBox:~$ ls
Desktop  Downloads  Music  Pictures  Templates
Documents  examples.desktop  newdirectory  Public  Videos
home@VirtualBox:~$
```

Figura 21. Comando mv

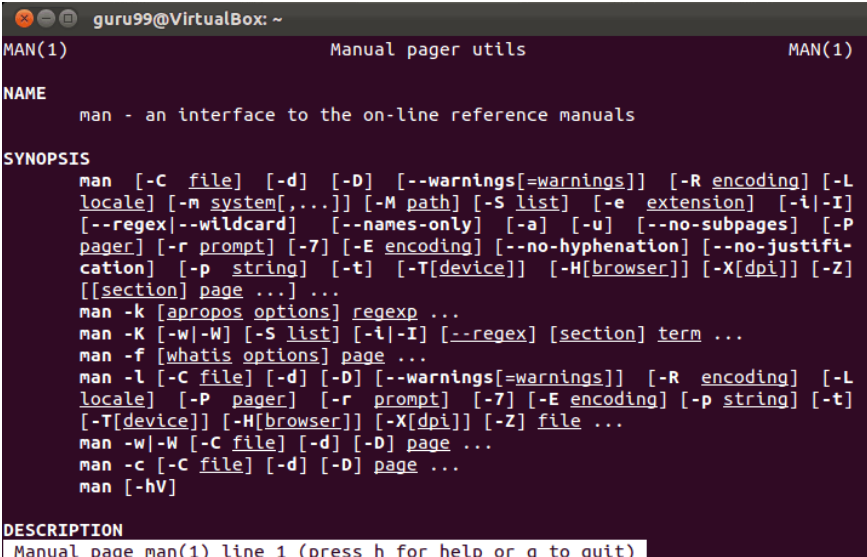
El comando 'man'

Man significa manual, que es un libro de referencia de un sistema operativo Linux . Es similar al archivo HELP que se encuentra en un software popular.

Para obtener ayuda sobre cualquier comando que no comprenda, puede escribir **man**

La terminal abriría la página del manual para ese comando.

Por ejemplo, si escribimos **man man** y presionamos enter; terminal nos daría información sobre el comando man.



```
MAN(1) Manual pager utils MAN(1)
NAME
  man - an interface to the on-line reference manuals
SYNOPSIS
  man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
  locale] [-m system,...]] [-M path] [-S list] [-e extension] [-i|-I]
  [--regex|--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P
  pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
  cation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
  [[section] page ...] ...
  man -k [apropos options] regexp ...
  man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
  man -f [whatis options] page ...
  man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
  locale] [-P pager] [-r prompt] [-7] [-E encoding] [-p string] [-t]
  [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
  man -w|-W [-C file] [-d] [-D] page ...
  man -c [-C file] [-d] [-D] page ...
  man [-hv]
DESCRIPTION
  Manual page man(1) line 1 (press h for help or q to quit)
```

Figura 22. Despliegue del comando man

El comando de history

El comando history muestra todos los comandos básicos en Linux que ha utilizado en el pasado para la sesión de terminal actual. Esto puede ayudarlo a consultar los comandos antiguos que ha ingresado y volver a utilizarlos en sus operaciones.

```
guru99@VirtualBox:~$ history
1  cat > sample
2  cat sample
3  cat sample ^a
4  cat sample a
5  cat sample | grep a
6  cat sample | grep ^a
7  useradd home
8  useradd mycomputer
9  sudo useradd mycomputer
10 sudo adduser MyLinux
11 sudo adduser mylinux
12 vi scriptsample.sh
```

Figura 23. Despliegue del comando history

El comando clear

Este comando despeja todo el desorden en la terminal y le brinda una ventana limpia en la que trabajar, al igual que cuando inicia la terminal.

```
141 man
142 3a
143 man intro
144 man ls
145 man cat
146 man man
147 history
148 146
149 history 146
150 history
151 clear
152 history
guru99@VirtualBox:~$ clear
The window gets cleared
guru99@VirtualBox:~$
```

Figura 24. Despliegue del comando clear

comando 'pr'

Este comando ayuda a formatear el archivo para imprimir en el terminal. Hay muchas opciones disponibles con este comando que ayudan a realizar los cambios de formato deseados en el archivo. Las opciones 'pr' más utilizadas se enumeran a continuación.

Tabla 2:

Opciones del comando pr

Opción	Función
-X	Divide los datos en columnas 'x'
-h "encabezado"	Asigna el valor de "encabezado" como encabezado del informe
-t	No imprime el encabezado y los márgenes superior / inferior
-D	Espacio doble en el archivo de salida
-norte	Denota todas las líneas con números
-l longitud de la página	Define las líneas (longitud de la página) en una página. El valor predeterminado es 56
-o margen	Formatea la página por el número de margen

División de datos en columnas

' Tools' es un archivo (que se muestra a continuación).

```
home@VirtualBox:~$ cat Tools
5/16" - 3/4" Standard Depth (6 Point)
3/8" - 3/4" Deep (6 Point)
9mm - 19mm Standard Depth (6 Point)
9mm - 19mm Deep (6 Point)
Ratchet
Extension - 3",6",12",18"
Universal Joint
Fractional Universal Impact Socket Set 3/8" - 3/4"
Metric Universal Impact Socket Set 9mm - 19mm
Slip Joint 6"
Needle Nose 6"
Diagonal Cutter 7"
Channel Locks 12" (water pump)
Long Reach End Cutter (Channel Lock #748)
Vise Grip Pliers 10" (10WR)
home@VirtualBox:~$
```

Figura 25. Despliegue del comando cat Tools

Queremos que su contenido se organice en tres columnas. La sintaxis para el mismo sería: `pr -x Nombre de archivo`

La opción '-x' con el comando 'pr' divide los datos en x columnas.


```
home@VirtualBox:~$ pr -3 Tools
```

Tools		
5/16" - 3/4" Standa	Extension - 3",6",1	Needle Nose 6"
3/8" - 3/4" Deep (6	Universal Joint	Diagonal Cutter 7"
9mm - 19mm Standard	Fractional Universa	Channel Locks 12" (
9mm - 19mm Deep (6	Metric Universal Im	Long Reach End Cutt
Ratchet	Slip Joint 6"	Vise Grip Pliers 10

Figura 26. Despliegue del comando pr -3 Tools

Asignar un encabezado

La sintaxis es: `pr -h "Encabezado" Nombre de archivo`

Las opciones '-h' asignan el valor de "encabezado" como encabezado del informe.

```
home@VirtualBox:~$ pr -3 -h "Important Tools" Tools
```

Important Tools		
5/16" - 3/4" Standa	Extension - 3",6",1	Needle Nose 6"
3/8" - 3/4" Deep (6	Universal Joint	Diagonal Cutter 7"
9mm - 19mm Standard	Fractional Universa	Channel Locks 12" (
9mm - 19mm Deep (6	Metric Universal Im	Long Reach End Cutt
Ratchet	Slip Joint 6"	Vise Grip Pliers 10

Figura 27. Despliegue del comando pr -3 -h

Como se muestra arriba, hemos organizado el archivo en 3 columnas y le hemos asignado un encabezado.

Imprimir un archivo

Una vez que haya terminado con el formateo y sea el momento de obtener una copia impresa del archivo, debe usar el siguiente comando: `lp`

`Nombre_de_archivo` o `lpr Nombre_de_archivo`

En caso de que desee imprimir varias copias del archivo, puede utilizar el modificador de número.

Print 10 copies of a File

Specify -nNumber with 'lp' command

```
guru99@VirtualBox:~$ lp -n10 testfile
```

Specify Number with lpr command

```
guru99@VirtualBox:~$ lpr 10 testfile
```

Figura 28. Despliegue del comando lp

En caso de que tenga varias impresoras configuradas, puede especificar una impresora en particular usando el modificador Impresora

In case of multiple printers , specify a particular printer

-dPrinter with lp command

```
guru99@VirtualBox:~$ lp -dHPofficejet testfile
```

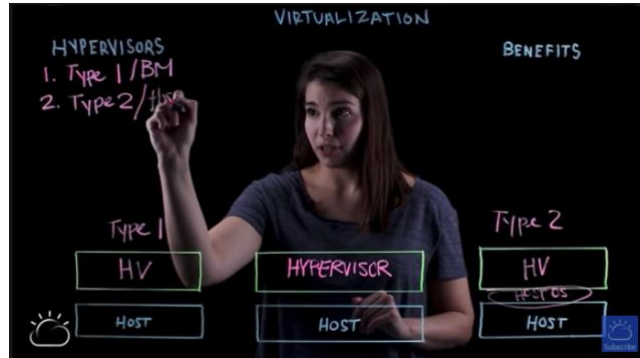
-Pprinter with lpr command

```
guru99@VirtualBox:~$ lpr -PHPofficejet testfile
```

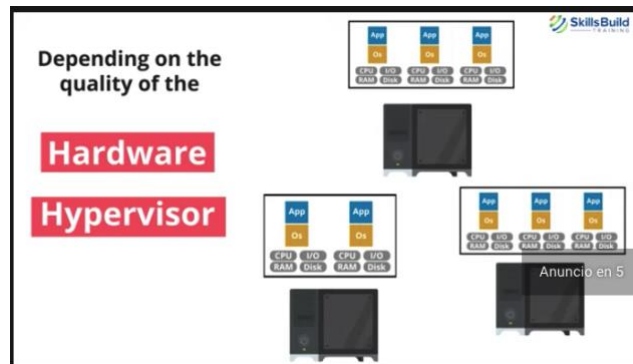
Figura 29. Despliegue del comando lp seleccionando diferentes impresoras

Recursos complementarios

- Video sobre Virtualización y comandos Linux



<https://www.youtube.com/watch?v=FZR0rG3HKIk>



<https://www.youtube.com/watch?v=dhQtoYEODIk>

```
The directory tree looks like:
└─ home
   └─ lanes
      └─ kenlm
         ├── builder
         ├── filter
         ├── interpolate
         ├── neural
         └─ wrappers
            └─ util
               ├── double-conversion
               └─ stream

$ cd /home/lanes/kenlm/util/stream
$ cd ../double-conversion/../../lm/.wrappers/./builder/
```

<https://www.youtube.com/watch?v=2PGnYjbYuUo>

Bibliografía

Tanenbaum, Andrew S. Sistemas operativos modernos. Pearson Educación, 2003.

Stallings, W., Aguilar, L. J., Dodero, J. M., Torres, E., & Mora, M. K. (1997). Sistemas operativos (Vol. 732). Prentice Hall.

Portnoy, M. (2012). Virtualization essentials (Vol. 19). John Wiley & Sons.