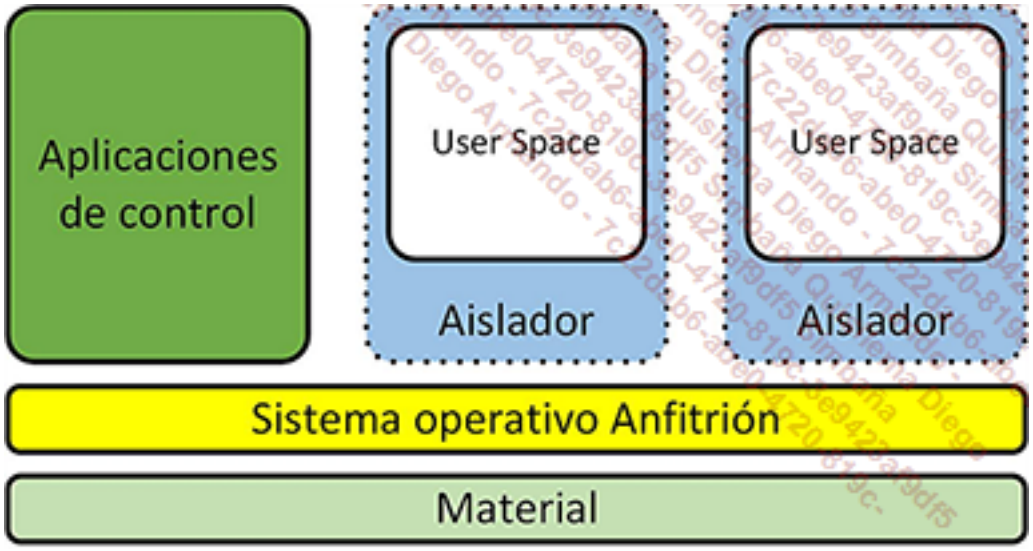
Métodos de virtualización

1. El aislamiento

El aislamiento consiste en ejecutar aplicaciones en entornos aislados los unos de los otros, llamados contextos o zonas de aislamiento. Algunas instancias de una misma aplicación se pueden arrancar simultáneamente, cada una en su contexto, incluso si esta aplicación no ha sido concebida con ese propósito. La aplicación no corre en una verdadera máquina virtual, sino que se encuentra aislada con respecto de las otras gracias a una funcionalidad del sistema operativo. El aislador más conocido en Unix es **chroot**, que permite un cambio de raíz de sistema.



Principio de los aisladores

El principio de chroot es simple. Un directorio Unix contiene todo lo necesario para ejecutar un programa, por ejemplo, un servidor web o ftp. En este directorio se encontrará la aplicación así como sus archivos, las bibliotecas y los comandos que esta utilizará. Una vez que está en su sitio, se aplica el comando chroot a ese directorio, el cual se convierte en la raíz. Los procesos del servidor solamente tienen acceso a los datos que se encuentran en esta nueva raíz y no puede salir de ella.

El núcleo de Linux presenta mecanismos de aislamiento integrados que se llaman **namespaces** o "espacios de nombre " que definen lo que los procesos podrán ver. El

principio es prácticamente el mismo que para chroot (un punto de montaje o directorio contiene todos los elementos necesarios para ejecutar la aplicación). Todos los procesos que se encuentren en el mismo es namespace podrán comunicar entre ellos pero no con otros que se encuentren fuera del namespace. Una particularidad interesante es que el primer proceso tiene como número identificador, o PID, el 1 en el namespace, un PID ordinario desde el punto de vista del anfitrión. Existen distintos tipos de namespaces que permiten aislar procesos, almacenaje, redes, usuarios y grupos, etc.

Otro mecanismo integrado en el núcleo, los **cgroups** o grupos de control, permite controlar el uso de los recursos hecho por uno o algunos procesos. De esta manera se puede limitar el consumo CPU o de memoria de uno o varios procesos y aplicar este control en un namespace.

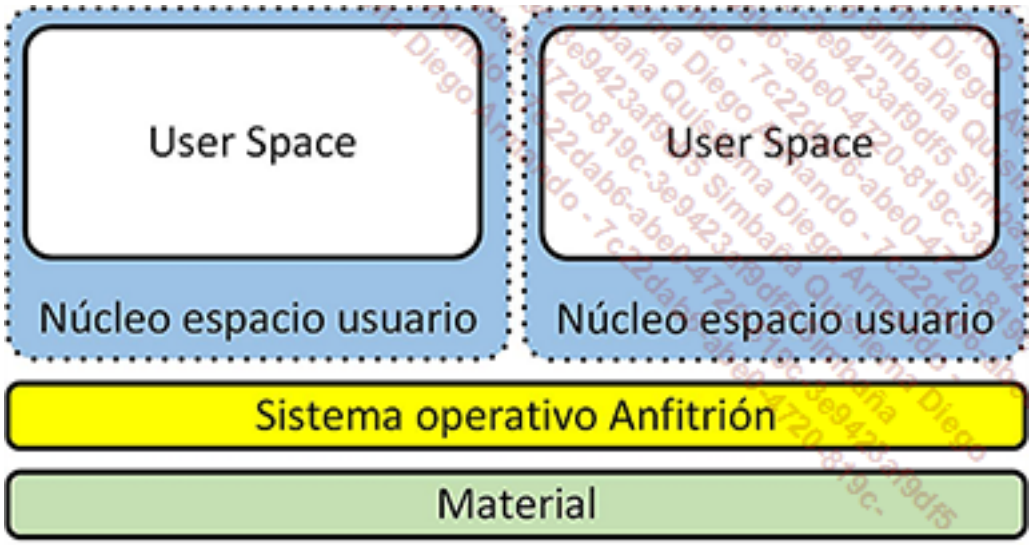
Los contenedores (**containers**), actualmente la última solución de aislamiento en Linux, son la asociación de los namespaces y los cgroups. Su uso fue bastante complicado hasta la aparición de herramientas para simplificar su creación y despliegue. Al principio fue LXC, pero sobre todo **Docker**.

En este sentido, la aparición de Docker ha sido una revolución en el aislamiento en Linux. Docker se estudiará con detalle en este mismo capítulo.

El aislamiento funciona de igual manera en una máquina virtual o física.

2. Núcleo en el espacio usuario

Un núcleo, en principio, no ha sido concebido para funcionar en el espacio de usuario. Sin embargo, es posible adaptar algunos de ellos en ese sentido, incluido el de Linux. Cada núcleo ejecutado dispondrá de su propio espacio usuario, y cada aplicación ejecutada en ese espacio solamente verá ese núcleo.

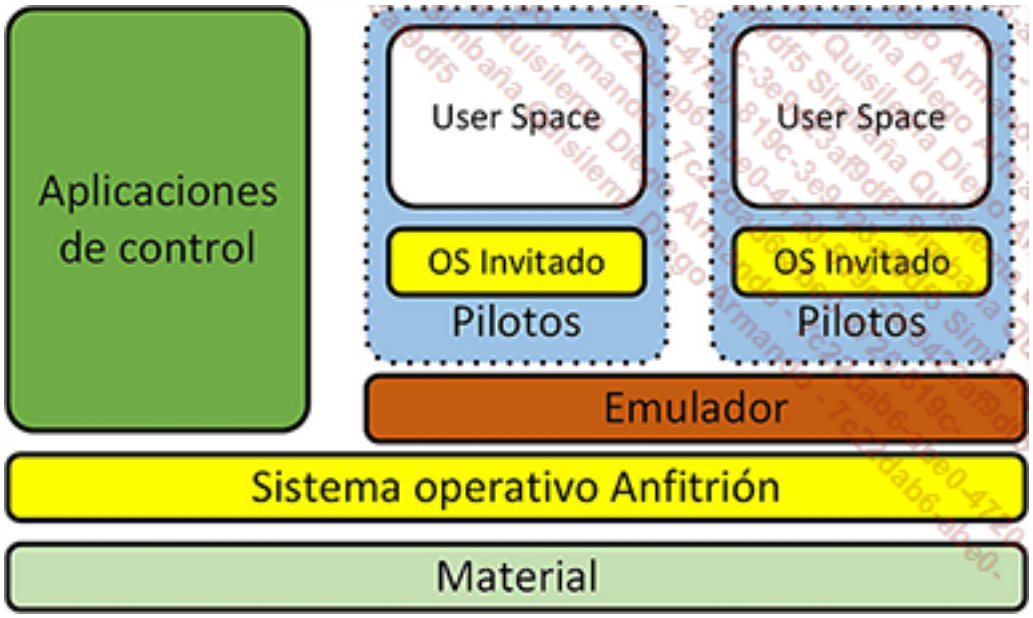


Principio del núcleo en el espacio usuario

Esta solución que parece atractiva, en realidad, no lo es tanto. Se trata de un apilamiento de núcleos, totalmente independientes del núcleo de base. Los proyectos User Mode Linux, Cooperative Linux o Adeos, los cuales estaban destinados a la experimentación y al desarrollo de los núcleos, parece que no han sido mantenidos.

3. Hipervisor de tipo 2

El hipervisor de tipo 2 es un programa especializado encargado de hacer funcionar sistemas operativos invitados (guests) en un sistema anfitrión (host) virtualizando o emulando el material dedicado a los sistemas invitados. Estos últimos solamente tienen acceso y dialogan con este material.



Virtualización por emulación

En este caso se habla de Full Virtualization. La máquina virtual se considera, a menudo, como un emulador y en realidad casi siempre es el caso porque a menudo el material tiene que ser emulado, haciendo bajar el rendimiento. Sin embargo, algunas máquinas virtuales utilizan, a veces, técnicas que permiten acelerar el funcionamiento del sistema invitado, sobre todo en el caso en que la arquitectura anfitrión/invitado sea idéntica.

Los programas que gestionan las máquinas virtuales ya sean de tipo 1 ó 2, se llaman VMM (*Virtual Machine Monitor*).

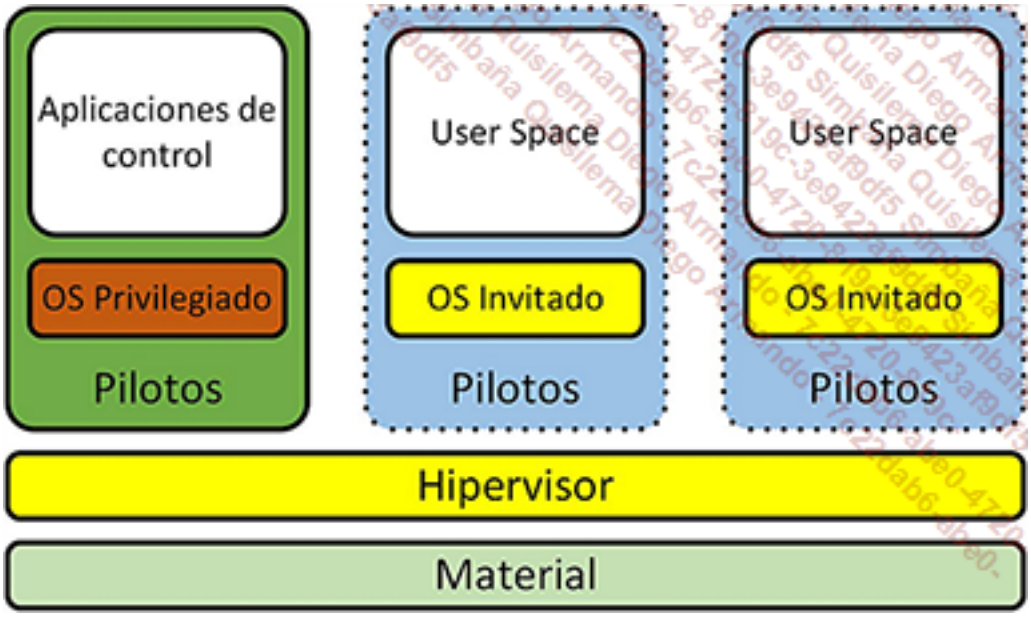
La mayoría de las soluciones de virtualización dirigidas al "gran público" en Linux, Windows o macOS son hipervisores de tipo 2. Se instalan en el sistema operativo existente, lo que no impide que las máquinas virtuales dispongan de capacidades de aceleración importantes a través del uso de técnicas de paravirtualización cuando sea posible:

- VMWare Fusion, Player, Server, Workstation
- VirtualBox
- Parallels Desktop, Server
- Microsoft VirtualPC, VirtualServer
- QEMU y KVM
- BOSCH (se pronuncia BOX)

4. Hipervisor de tipo 1

Un hipervisor de tipo 1 es un sistema cuyo núcleo ha sido optimizado específicamente con el objetivo de gestionar la arquitectura de destino de los sistemas operativos invitados. el sistema anfitrión también está virtualizado. Es el método que mejor rendimiento presenta actualmente, pero aparte de algunas soluciones gratuitas como Xen, es un método bastante caro.

Si un sistema operativo invitado « sabe » que está instalado en un hipervisor, puede ser optimizado y aumentar de esta manera su rendimiento. Esto se hará mediante un núcleo particular o usando pilotos específicos, en este caso se habla de paravirtualización.



Hipervisor de tipo 1

Los productos Xen, VMWare ESX o Microsoft Hyper-V son hipervisores. Por abuso de lenguaje, se llama a menudo hipervisor a todo sistema anfitrión capaz de controlar el funcionamiento de sistemas invitados.

5. Virtualización material

Algunas máquinas pueden estar específicamente adaptadas a la virtualización. Los

procesos y los controladores dispondrán de un juego de instrucciones que permitirán la virtualización al más bajo nivel. Existen muchas arquitecturas de este tipo, pero solo nos detendremos en las que se proponen en los sistemas de 64 bits: AMD-V (pacifica) e Intel-VT (vanderpool).

La mayoría de los supervisores están optimizados para explotar instrucciones adicionales.