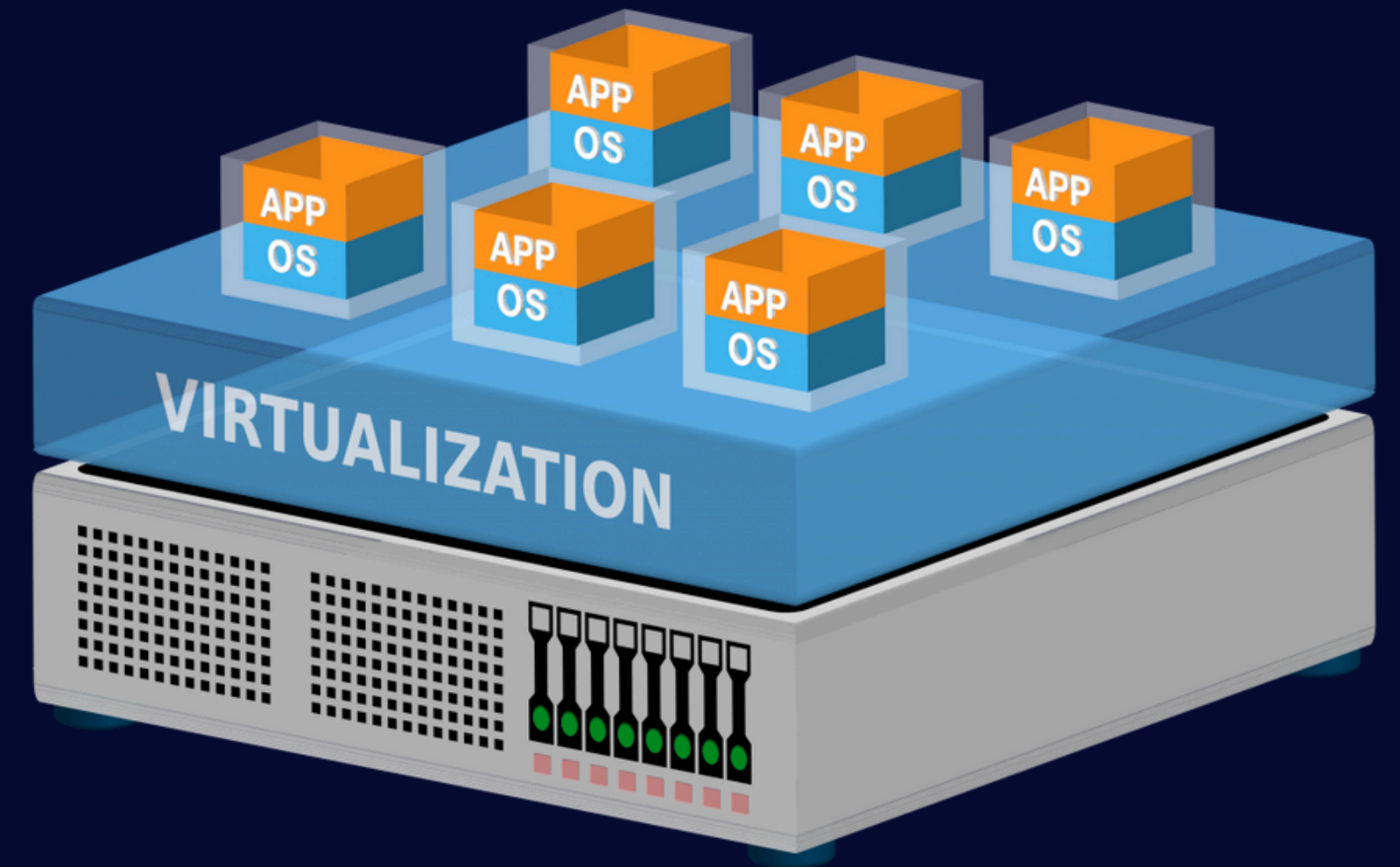


**Sistemas operativos**

# Ayudantía 9: Virtualización y precontrol

Profesores: Martín Gutiérrez, Víctor Reyes

Ayudantes: Diego Banda, Dante Hortuvia



# Contacto



dante.hortuvia@mail.udp.cl



doshuertos



[github.com/Doshuertos/Ayudantias\\_SO\\_CIT2010](https://github.com/Doshuertos/Ayudantias_SO_CIT2010)

## Sección 1

## Sección 2



diego.banda@mail.udp.cl

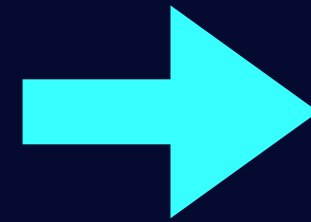
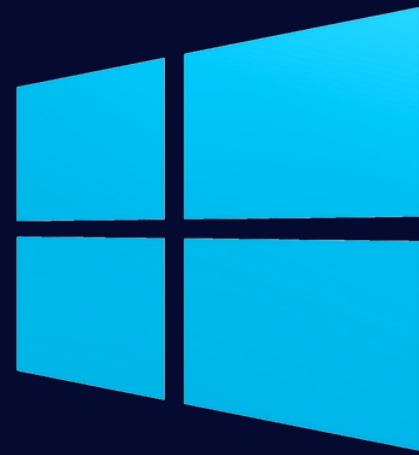


darkclouds



[github.com/DiegoBan/SO2025-I](https://github.com/DiegoBan/SO2025-I)

# Virtualización



etc...



Existen muchos Sistemas Operativos, si queremos utilizar más de uno por distintas decisiones, qué podemos hacer? → Virtualización

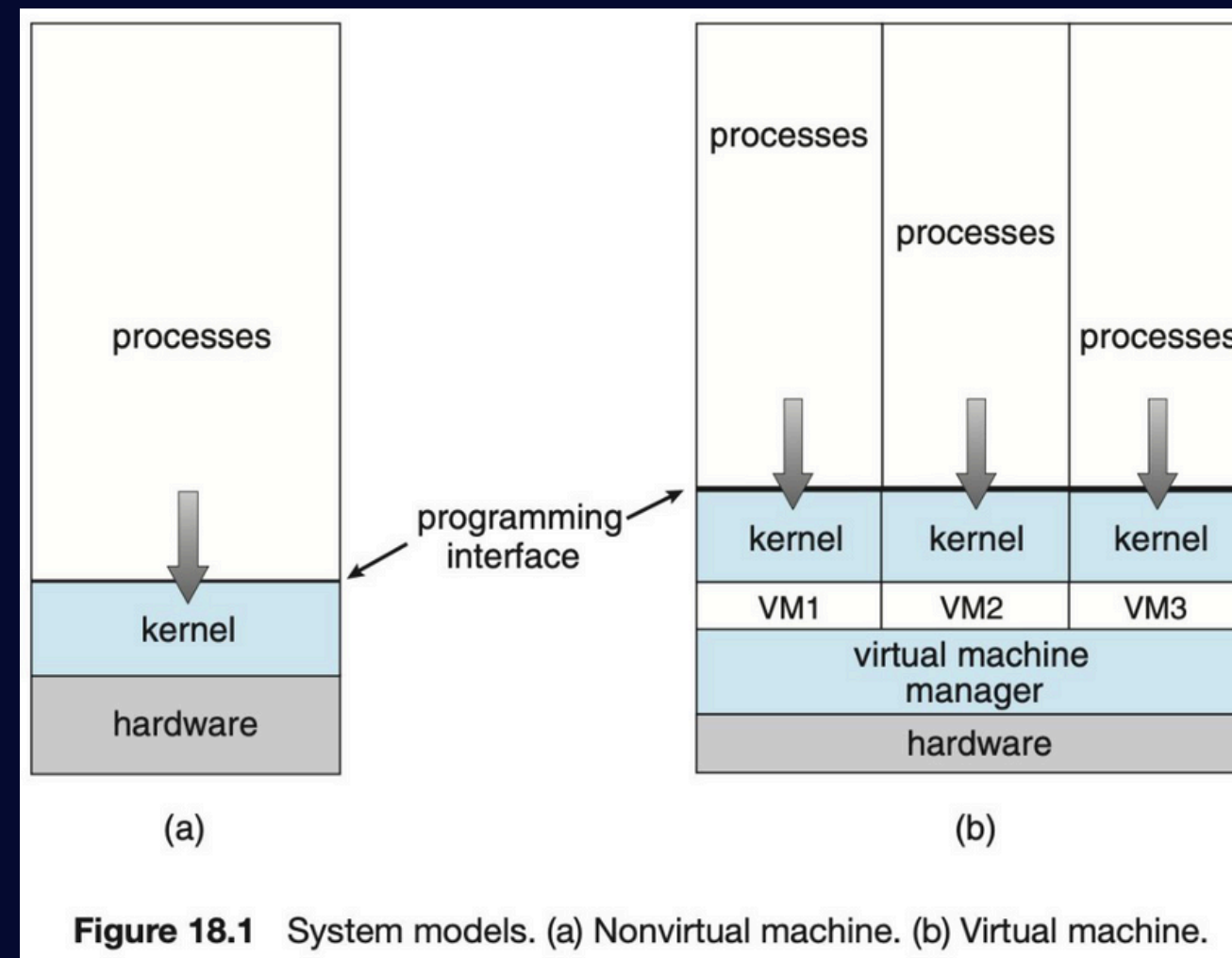
Como cuando quiero Windows para jugar jueguitos y Linux para trabajar y crearme hacker.



# Virtual Machines

- **Virtual Machines:** Maquinas virtuales de hardware que proveen una interfaz de los recursos del host al SO guest.

A partir de esto, se busca tener distintos SO coexistiendo en la misma maquina física (HW, un computador jeje).



# Virtual Machines

Con esto se busca:

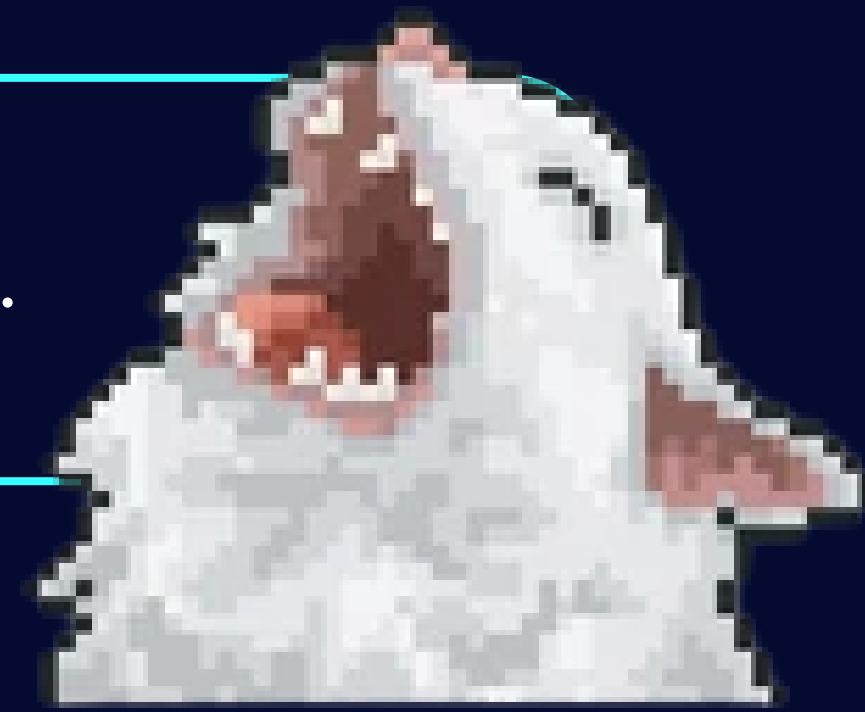
- Aislamiento entre los distintos sistemas involucrados.
- Manejo temporal de los SO.
- Concurrencia y diversidad de SO.
- Pruebas e investigación con SO.

Todo esto se maneja con:

- **Hipervisor:** Capa principal de VM, e interfaz hacia HW, es como la capa intermedia entre el SO Virtual y los componentes de nuestro computador.

Extra:

- SO host: Sistema operativo con hipervisor, que contiene el resto de SO's.
- SO guest: Sistema operativo virtualizado “dentro” de otro SO.



# Tipos de Hypervisor

## Tipo 0

Paravirtualización, todas las SO son concientes de el resto, la SO host ayuda a la virtualización.

## Tipo 1

Proveen entorno para tener VM's, a su vez, provee entorno para tener una máquina nativa.

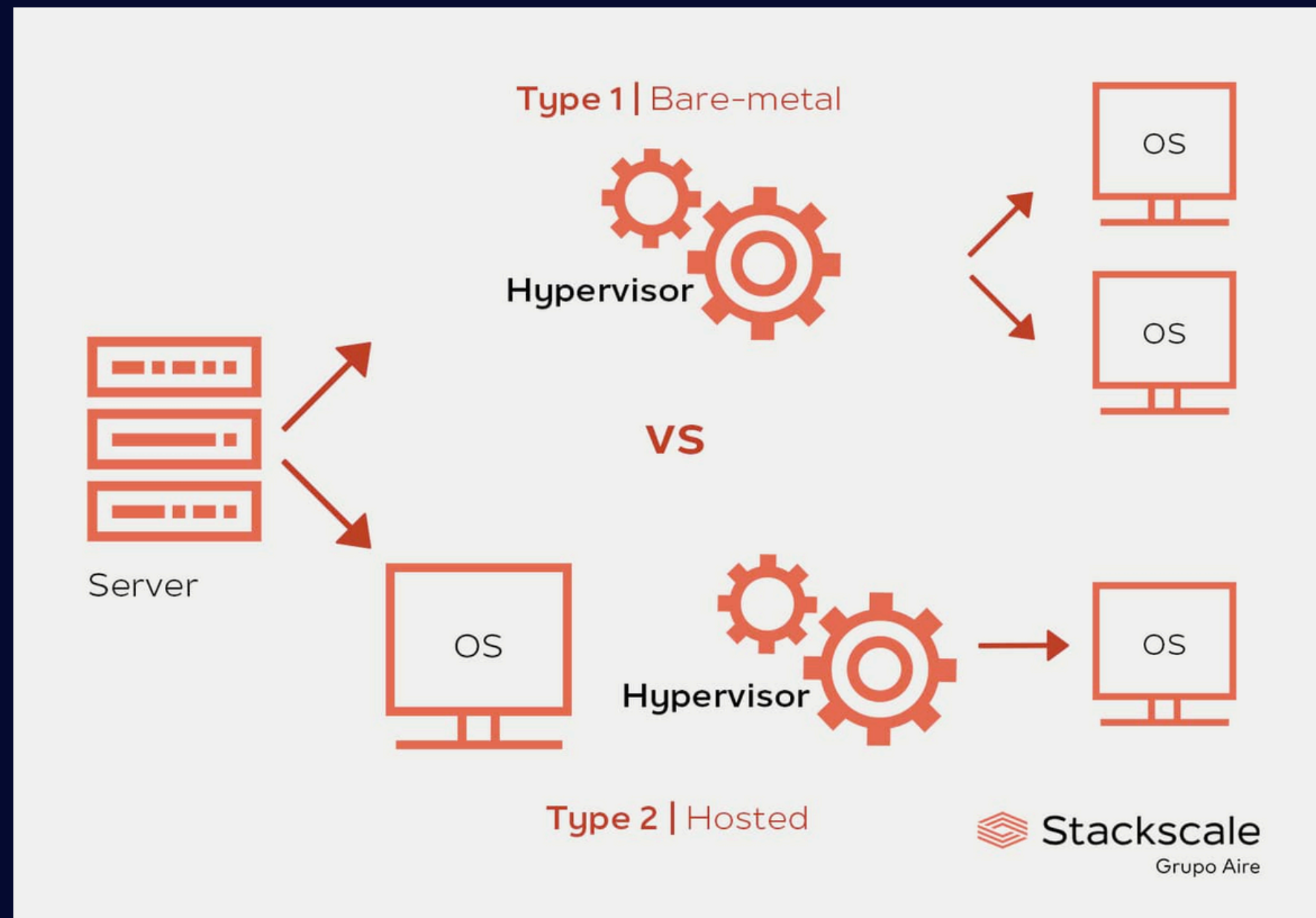
## Tipo 2

Desconocimiento total de la existencia de la virtualización, se trabaja a nivel de aplicación.





# Tipos de Hipervisor

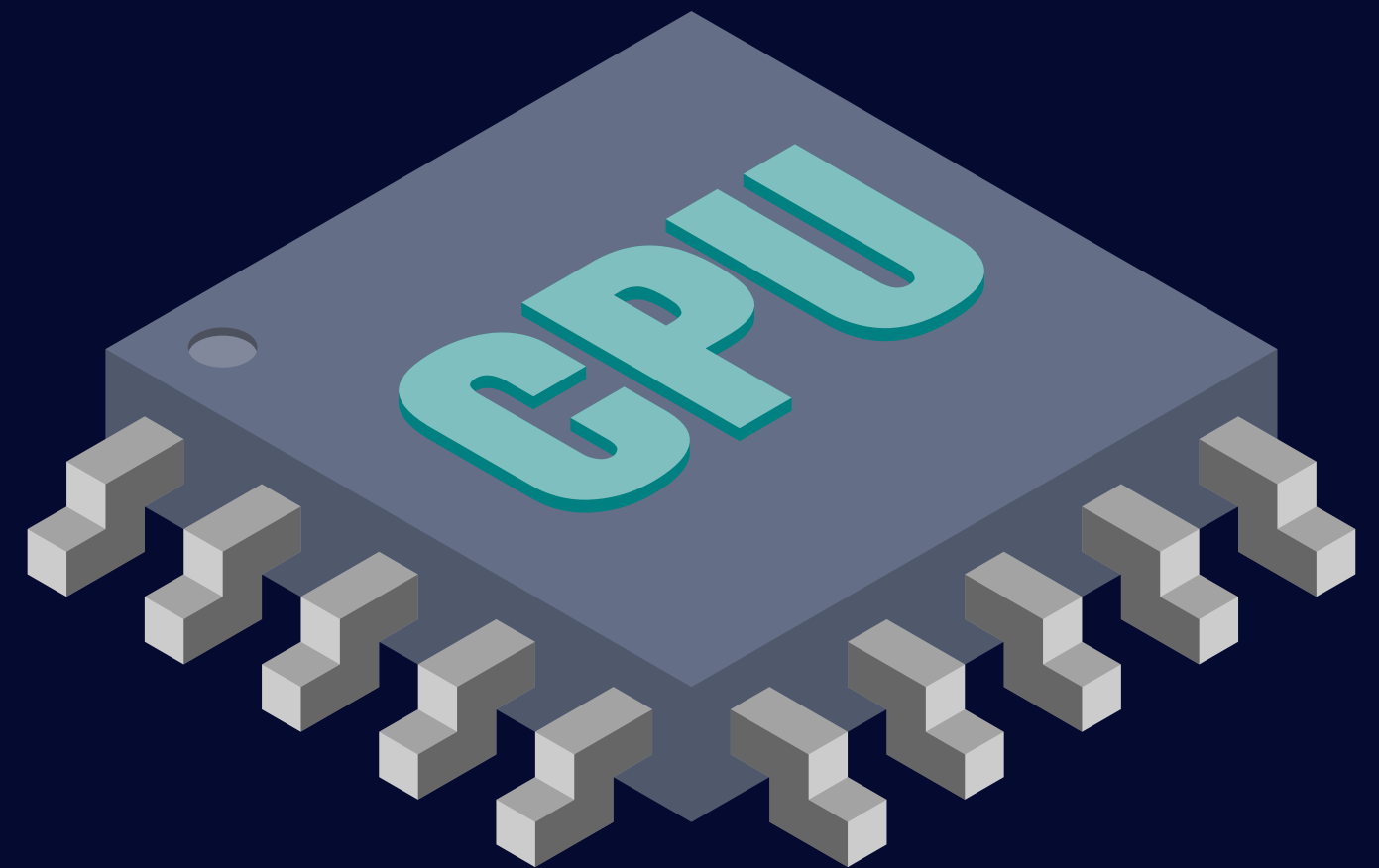


# Recursos: CPU

Ya tenemos distintos Sistemas Operativos, pero... Sigue siendo una máquina física.

Todo se sigue ejecutando a nivel físico en una sola CPU:

- **VCPU:** Virtual CPU (no procesa nada) es el estado que la VM huésped cree que tiene, para entregarle ahora sí, tiempo de CPU real.





# Syscalls el regreso?

Al no ser un SO “real”, sino que está virtualizado, no puede ejecutar syscalls llamando directamente al modo kernel, ya que en el SO host está en modo usuario, para ejecutar sus cosas hay dos formas...

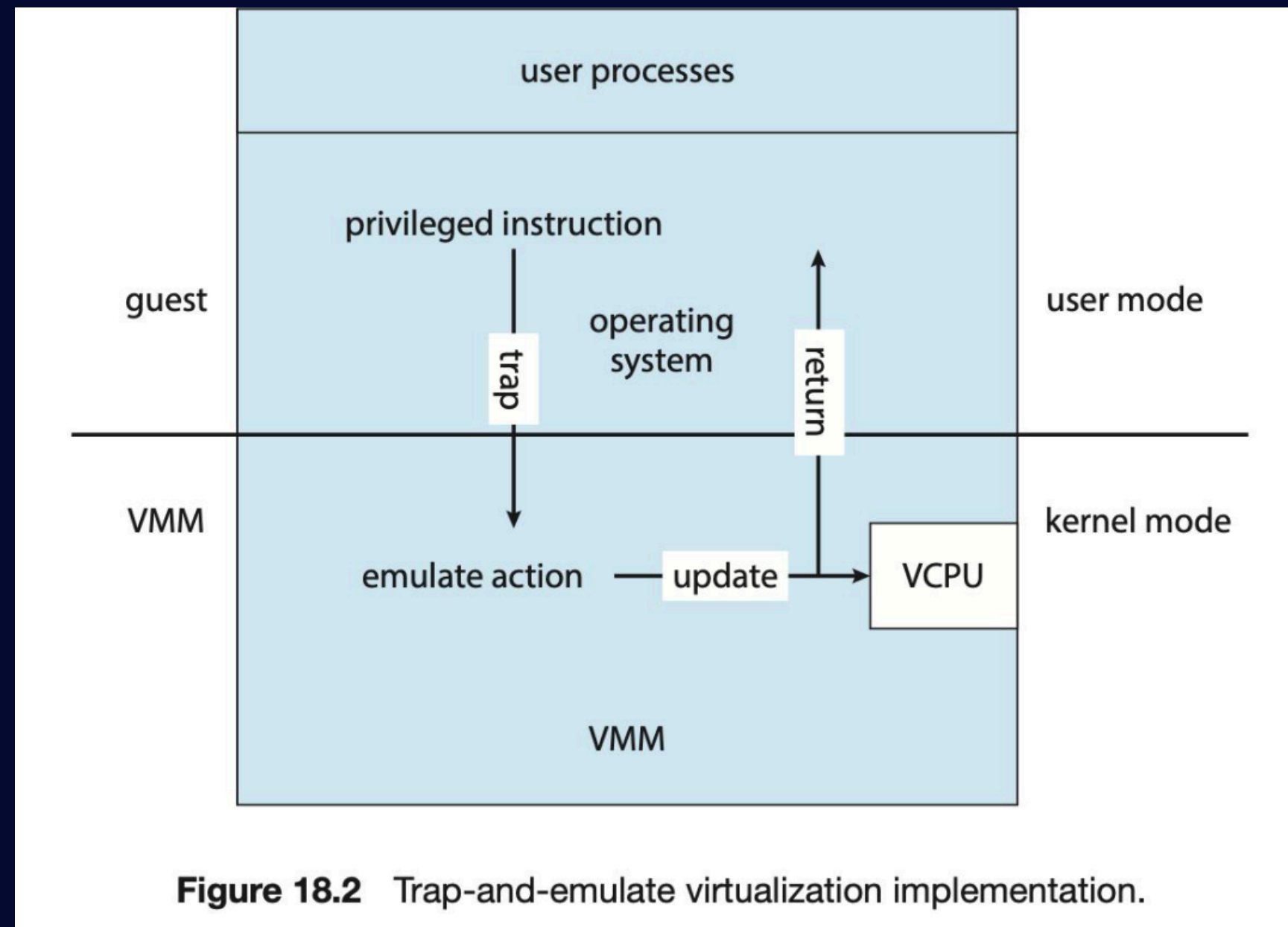
- **Trap and Emulate:**

1. Genera una instrucción que requiere de una Syscall.
2. Al estar en modo usuario e intentar algo sin los permisos adecuados: ERROR!!.
3. Producto del error (trap) se devuelve el control al VMM (hypervisor) y emula lo que intentó ejecutar.
4. Una vez ejecutado, se retorna control a SO guest.

Consecuencias: Instrucciones normales no demora, sin embargo las que van a modo kernel tardan más de lo normal.

# Syscalls el regreso?

- Trap and Emulate:



# Syscalls el regreso?

Al no ser un SO “real”, sino que está virtualizado, no puede ejecutar syscalls llamando directamente al modo kernel, ya que en el SO host está en modo usuario, para ejecutar sus cosas hay dos formas...

- **Binary Translation:**

- Si la VCPU está en modo usuario, se ejecutan normalmente.

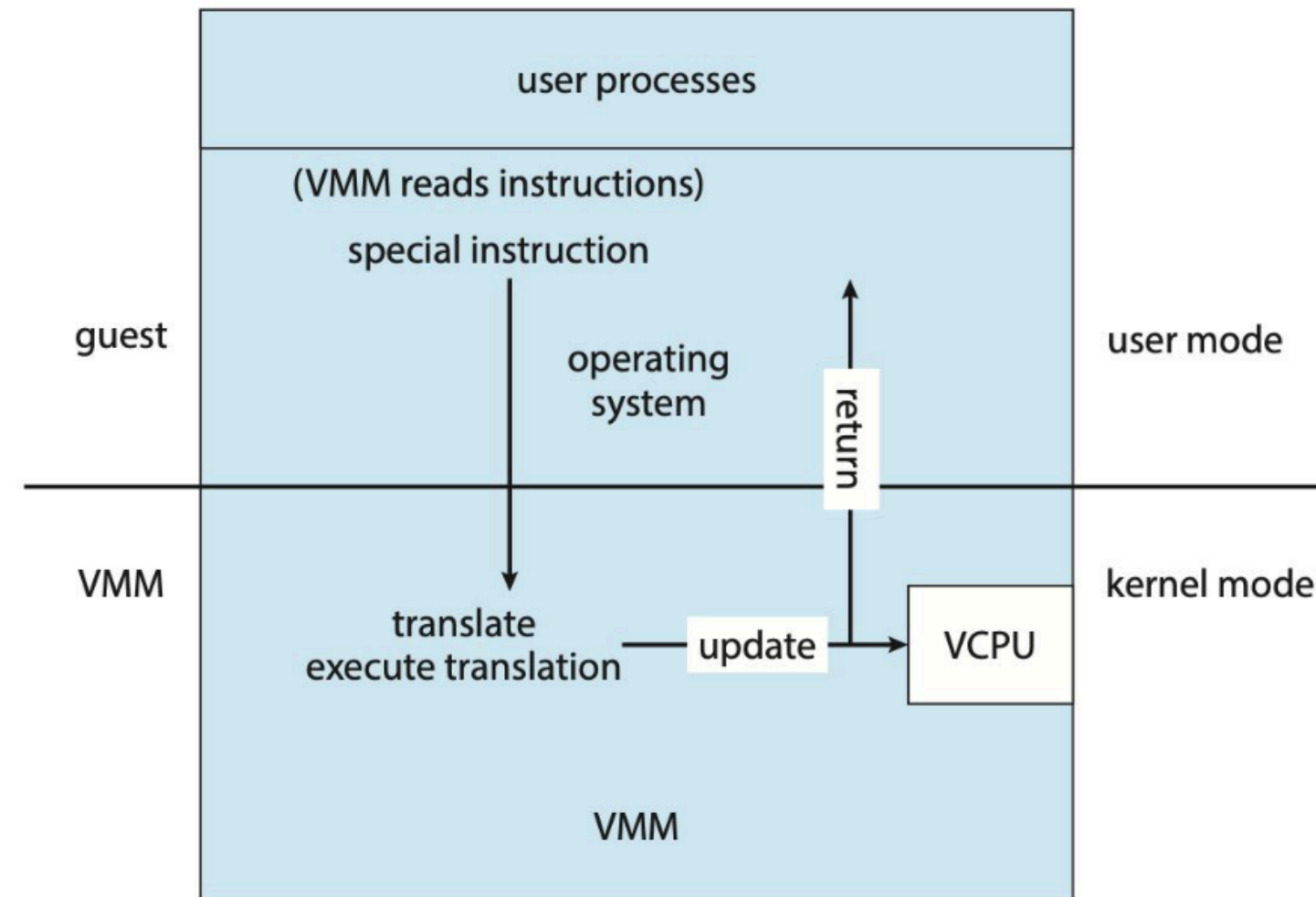
1. Si se quiere acceder al modo kernel, esa y las siguientes instrucciones relacionadas son analizadas por el VMM.

2. Se hace creer al SO guest que está en modo kernel, y se traducen las instrucciones por otras similares.

Consecuencias: Instrucciones normales no demora, sin embargo las que van a modo kernel tardan más de lo normal.

# Syscalls el regreso?

- Trap and Emulate:



**Figure 18.3** Binary translation virtualization implementation.

# Scheduling

En cuanto a scheduling también hay problemas... Y hay dos casos especiales para este tema:

1. En un caso en el que hay **suficientes CPU's** Físicas para suplir la demanda, trabajan como si estuvieran dedicadas a la VM.

En caso contrario, toca repartir el uso de CPU entre SO host, VM, VMM, etc...

2. Caso extremo: **Overcommitment** (sobre-compromiso), es cuando la o las VM's están configuradas de tal manera que esperan más CPU de la que existe realmente.

En estos casos, la VMM se encarga de agregar una capa extra de scheduling.

# RAM

Más y más problemas... En cuanto a memoria RAM también se debe de asignar correctamente....

Al igual que con la CPU, hay que tener un correcto orden de la memoria principal, puede que las VM's esperen más de la que realmente existe.

## **Tablas de paginación dobles**

Montan tablas de  
paginación sobre las  
tablas de paginación del  
SO host.

## **Balloon memory manager**

Pseudo-driver  
manejado por VMM  
que trabaja con las  
expectativas de uso de  
memoria de la VM.



# Ejercicios

Verdadero o Falso (Justifique las falsas).

- 1.\_\_\_\_ La asignación de espacio libre en un HDD tiene la característica de no ser escalable cuando se usa un bitmap
- 2.\_\_\_\_ Dos dispositivos distintos (por ejemplo, HDDs externos) se montan en dos puntos de montaje distintos.
- 3.\_\_\_\_ El hipervisor Tipo 2 se instala directamente sobre el hardware físico y no necesita un sistema operativo subyacente.
- 4.\_\_\_\_ En la virtualización, la caché no mejora el rendimiento de las máquinas virtuales.
- 5.\_\_\_\_ Es posible crear una máquina virtual dentro de otra máquina virtual



# Ejercicios

Responda las siguientes preguntas:

1. En el contexto de la memoria secundaria. ¿En qué se diferencia la asignación enlazada con la indexada?
2. ¿Cuál es la utilidad de un Virtual File System? Además, indique con qué elementos trabaja (al menos 2).
3. Describa las diferencias entre la virtualización mediante máquinas virtuales (VMs) y contenedores.
4. ¿Cuáles son las diferencias entre un hipervisor tipo 1 y un hipervisor tipo 2?
5. ¿Qué es la técnica de "Trap and Emulate" y cómo se implementa en la virtualización?



# Ejercicios

Responda las siguientes preguntas:

1. En un entorno donde se utilizan políticas de swap-out, ¿cómo optimizaría el sistema operativo el tiempo necesario para identificar y mover páginas menos relevantes al disco?
2. Explique cómo la asignación indexada multinivel optimiza el acceso aleatorio a los archivos y compare este enfoque con la asignación enlazada.
3. Explique cómo el concepto de "espera circular" puede eliminarse sin comprometer la funcionalidad de los procesos.
4. ¿Qué papel juega la tabla de montajes (mount table) en la virtualización de sistemas de archivos?
5. Dado un ejemplo de acceso a páginas, ¿cómo determinaría cuál política de reemplazo de páginas es más eficiente?

