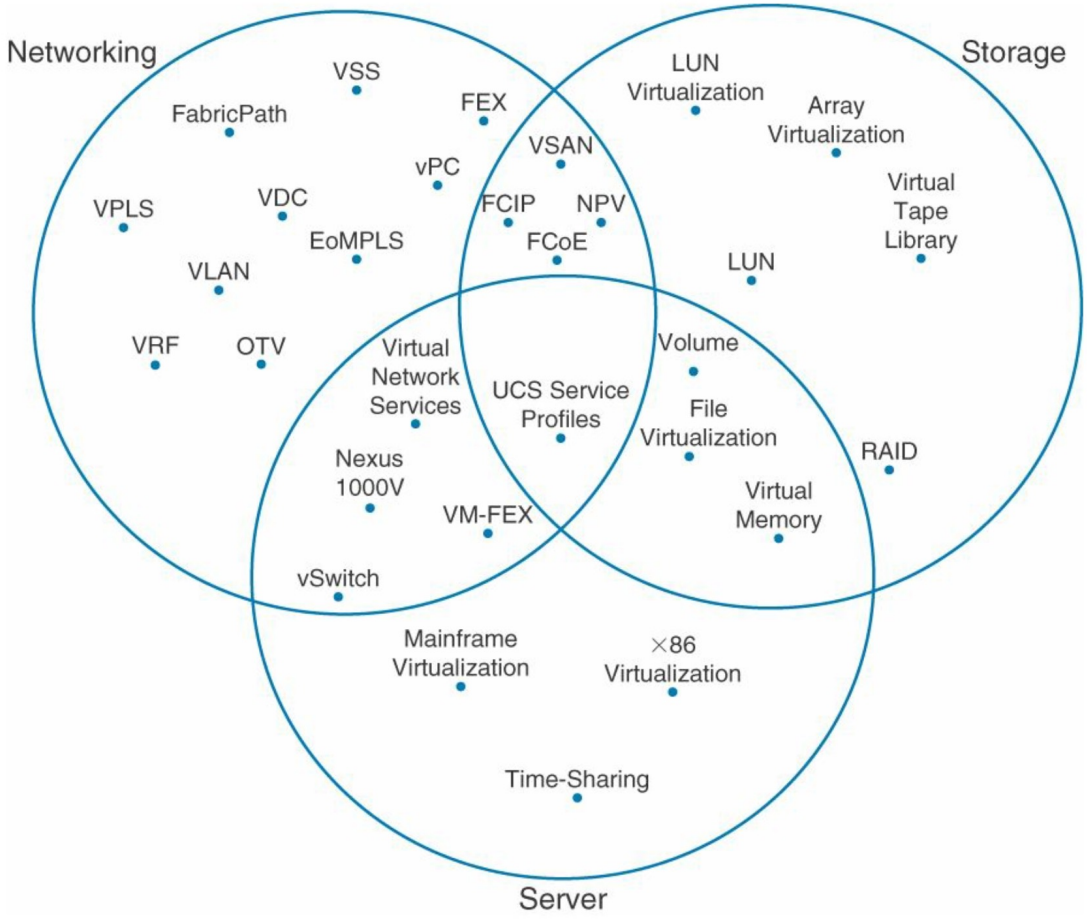
upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Redes de Nueva Generación
Área de Ingeniería Telemática

Virtualización

Virtualización, ¿dónde?

- Servidor
- Red
- Almacenamiento

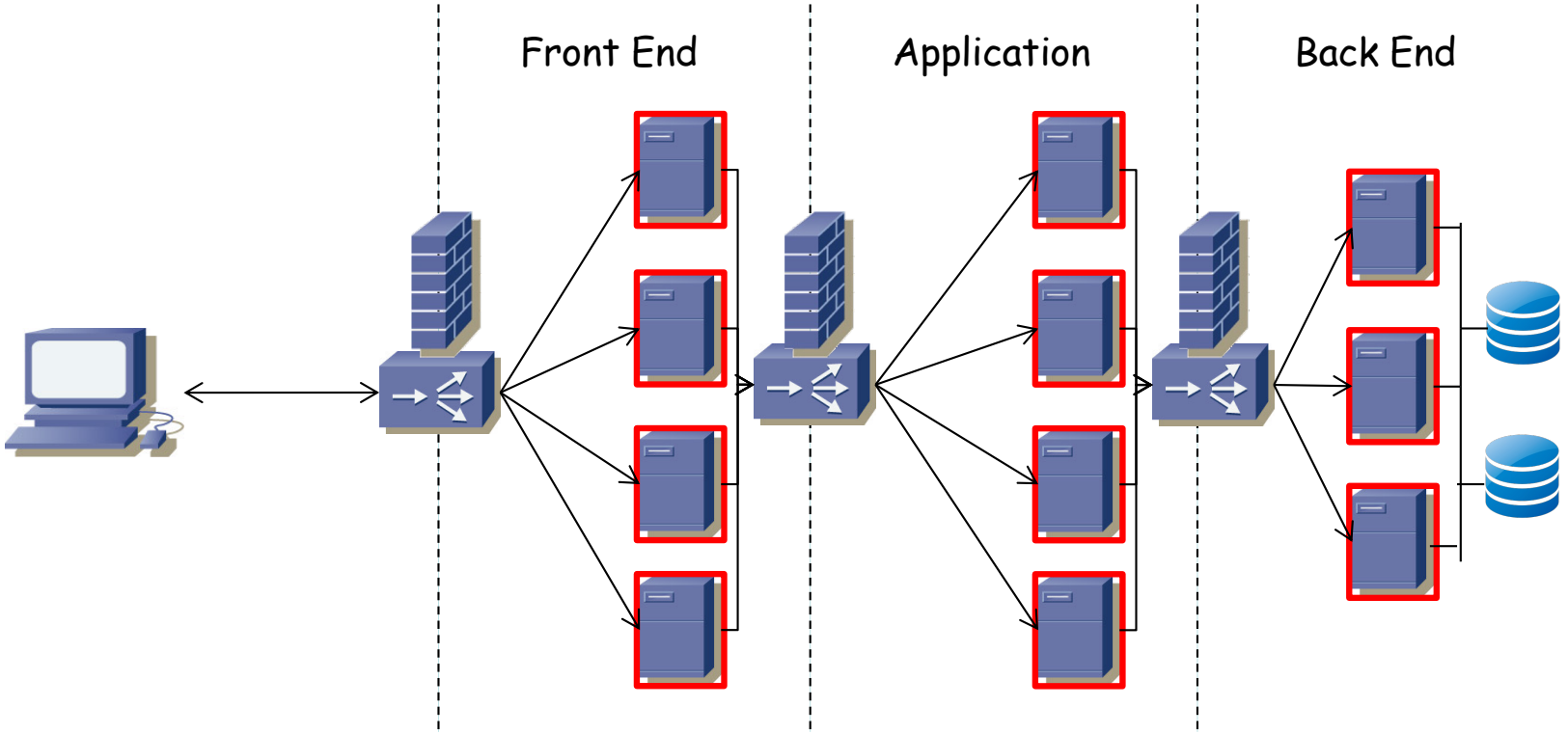


upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Virtualización de host

Servidores

- Host físicos
- A día de hoy existen más interfaces de red virtuales que físicos
- ¿De qué estamos hablando?

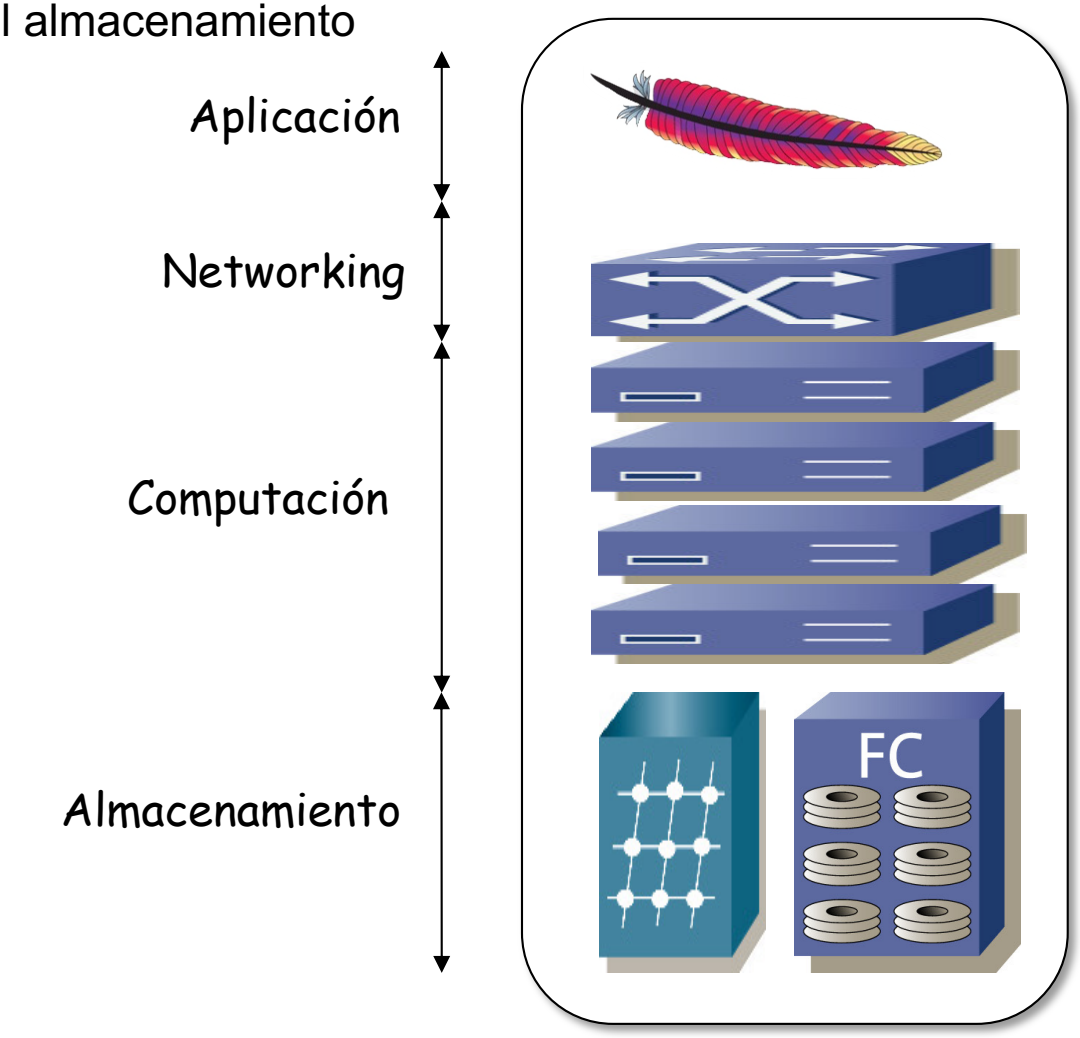


upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Application Silos

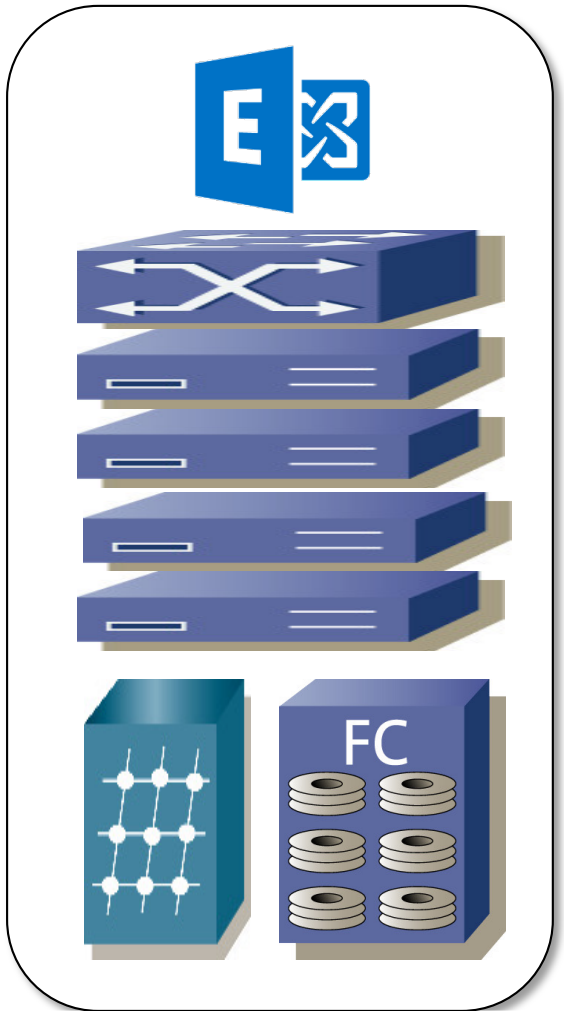
Application Silos

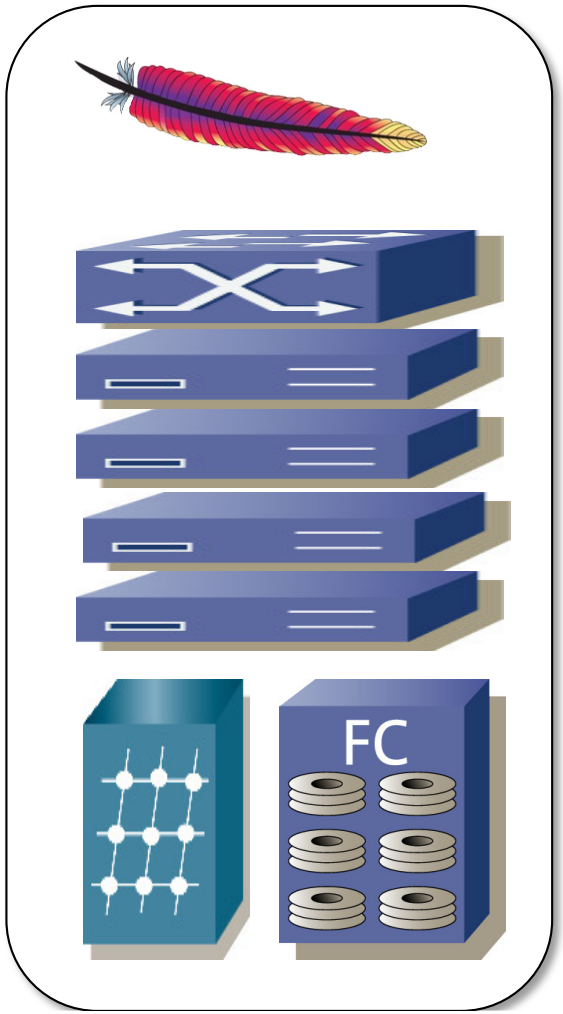
- En el entorno distribuido una nueva aplicación (software servidor) se desplegaba sobre un hardware independiente
- Una relación 1:1 entre la aplicación y el hardware servidor
- O como mucho 1:N porque tengamos múltiples servidores
- A esto habría que añadirle el almacenamiento
- Y la electrónica de red

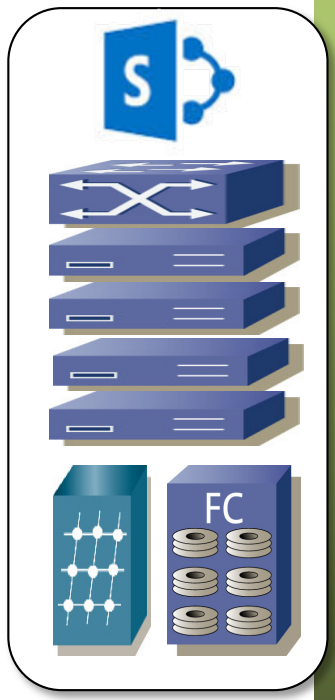


Application Silos

- ¿Y si tenemos otra aplicación?
- Cada una con sus servidores y almacenamiento
- El hardware no es reutilizable por otras aplicaciones

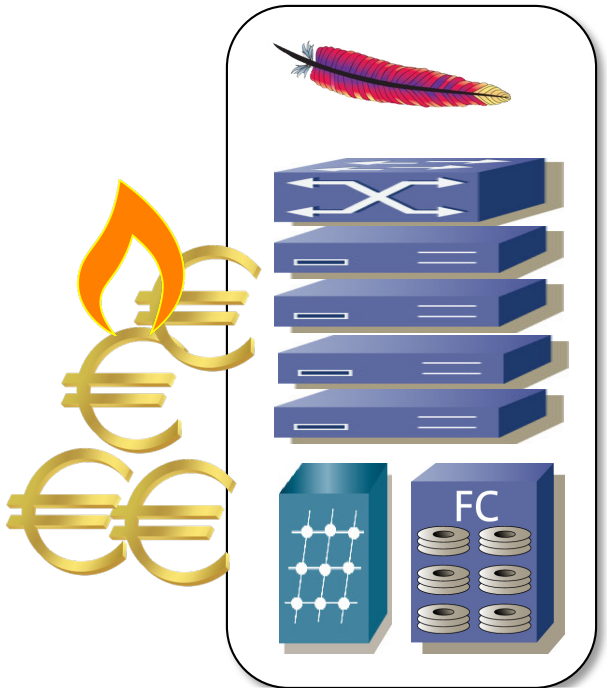






Application Silos

- La utilización (CPU) de los servidores es muy baja
- Esto es así para soportar incrementos de carga
- Si no es baja entonces ante un incremento de carga no es rápido provisionar nuevo hardware
- Lo mismo sucede con la utilización de los discos
- Esto se multiplica por el número de aplicaciones
- Pero ocupan todo el tiempo el espacio
- Y están encendidos, consumiendo potencia
- Y necesitando refrigeración
- Esto ha cambiado con la virtualización
- Consolidación

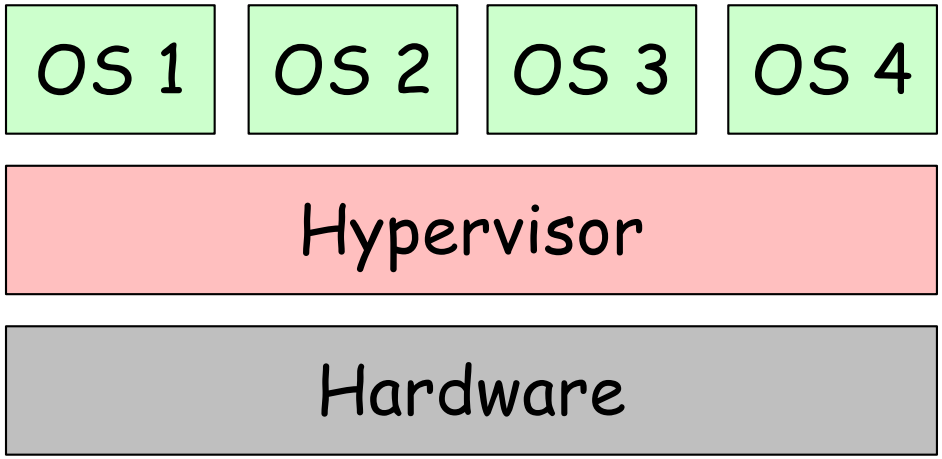


upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Tecnologías tras la virtualización de host

¿Virtualización?

- La idea básica de virtualización del host es bastante conocida
- Una capa software intermedia hace creer a un sistema operativo que tiene hardware dedicado
- En realidad esto lo hemos visto antes (…)

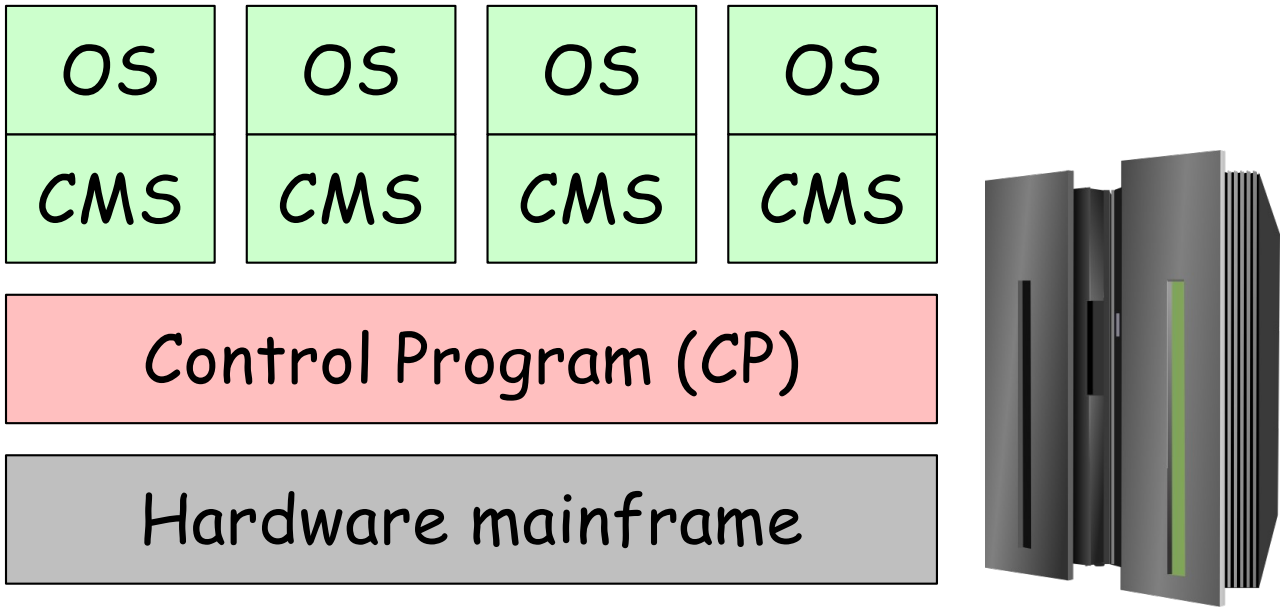


¿Virtualización?

- Es la misma idea detrás de las VLANs
- Los hosts de cada VLAN la ven como si estuvieran ellos solos en la LAN

¿Virtualización?

- En el entorno informático es un concepto muy antiguo
- A nivel de virtualización de sistemas operativos ya lo soportaban los mainframes en los 70s
- Comercialmente llega al entorno PC a principios de los 00s (VMware)

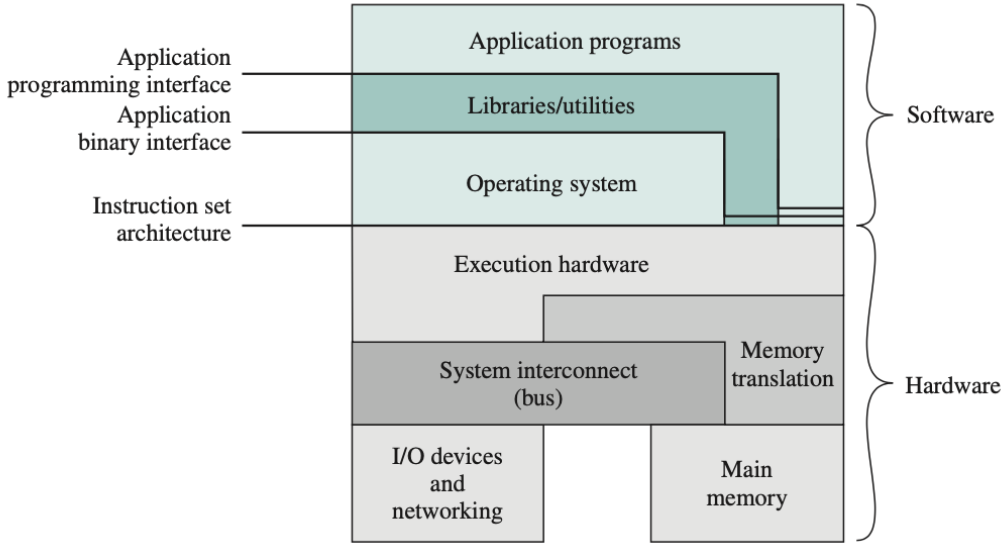


upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

Virtualización de memoria y multiproceso

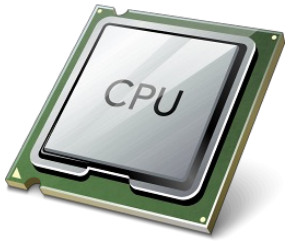
Kernel, Sistema Operativo

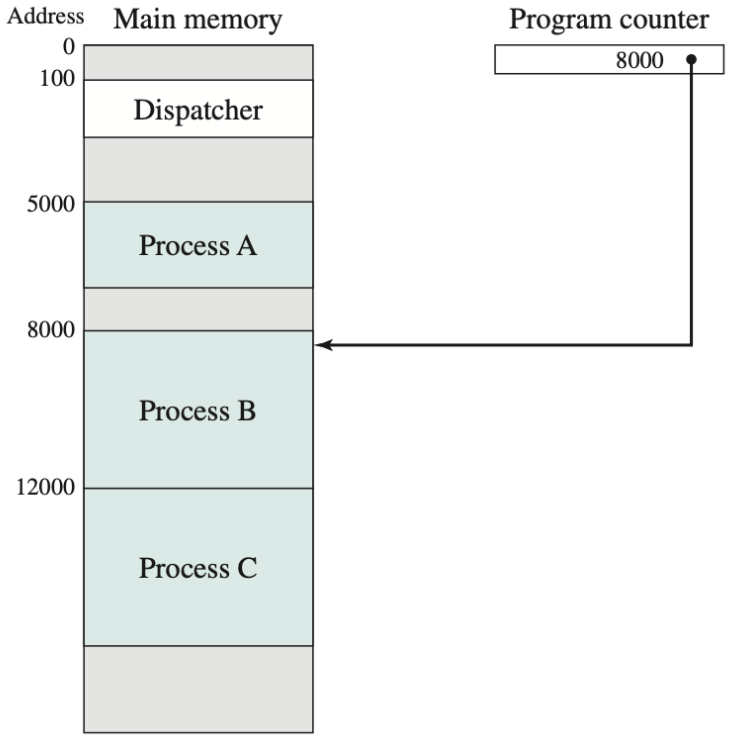
- No vamos a entrar a hablar de arquitecturas de kernels monolíticos, microkernels, servicios del S.O.
- Estamos tratando de un programa (o conjunto de programas)
- Controla el acceso de otros programas a los recursos hardware
- Oculta los detalles del hardware al software, proporcionando independencia de los mismos
- Proporciona planificación de procesos, control de acceso a memoria, acceso a dispositivos, etc



Procesos

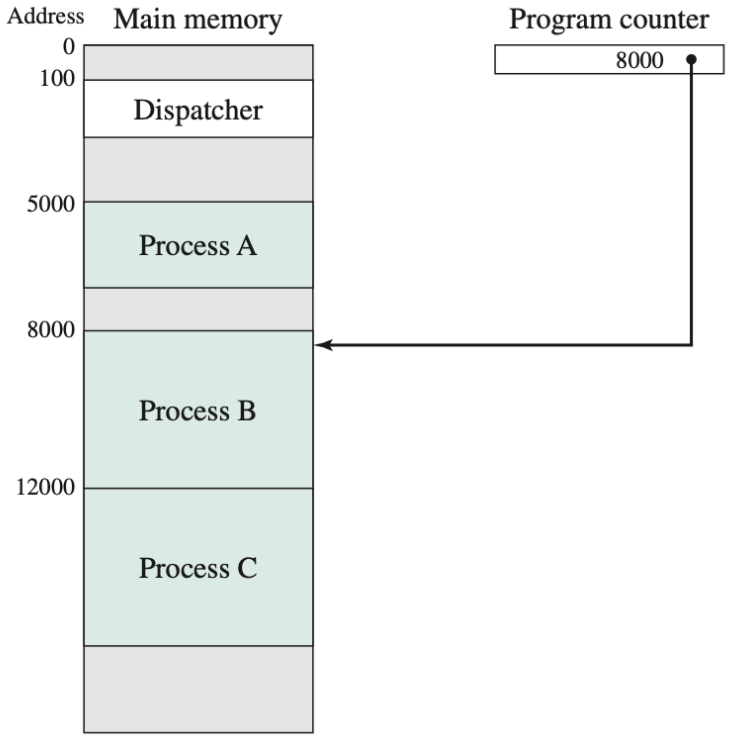
- El código del sistema operativo es ejecutado por la CPU
- Para que pueda ejecutarse el código de una aplicación de usuario el S.O. debe dejar de usar la CPU para que ejecute ese otro código
- El S.O. necesita mecanismos para recuperar el control de la CPU
- Es decir, para que deje de ejecutar el código del programa de usuario y vuelva al código del kernel





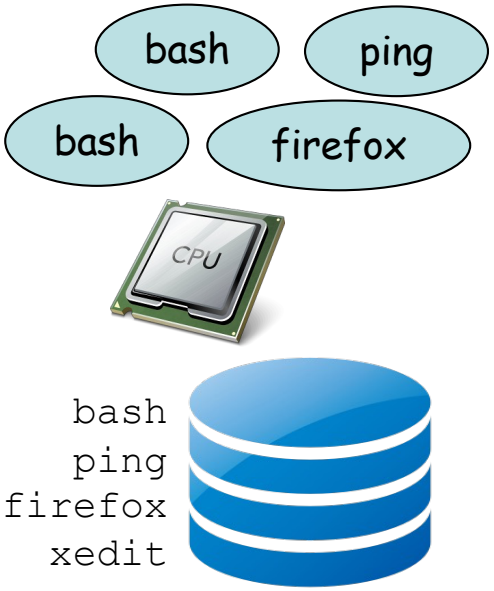
Procesos

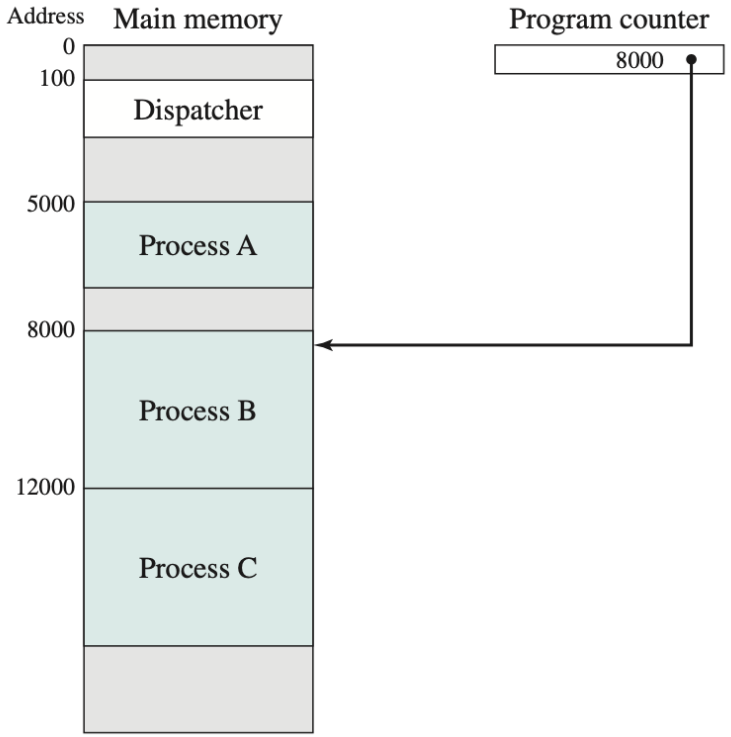
- Un proceso es un programa en ejecución
- /usr/bin/bash, /usr/bin/ping, /usr/bin/firefox, /usr/bin/xedit son ficheros en disco que contienen código ejecutable



Procesos

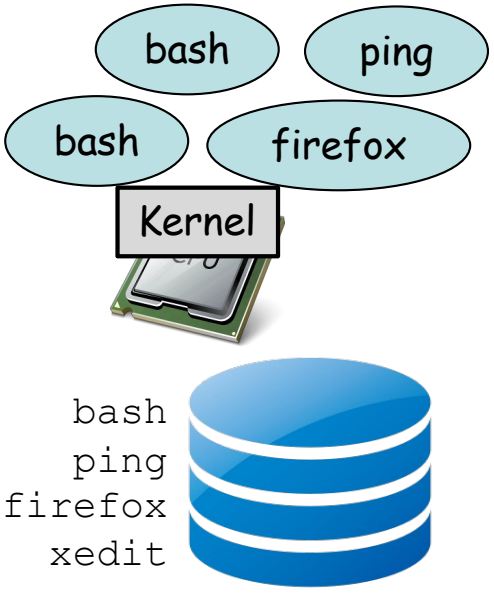
- Un proceso es un programa en ejecución
- /usr/bin/bash, /usr/bin/ping, /usr/bin/firefox, /usr/bin/xedit son ficheros en disco que contienen código ejecutable
- El Kernel crea una entidad de ejecución que llamamos ”proceso”
- Tiene un hilo de ejecución (thread) así como otras estructuras asociadas en el kernel (fds, memoria, etc)
- Podemos estar ejecutando el mismo programa en varios procesos

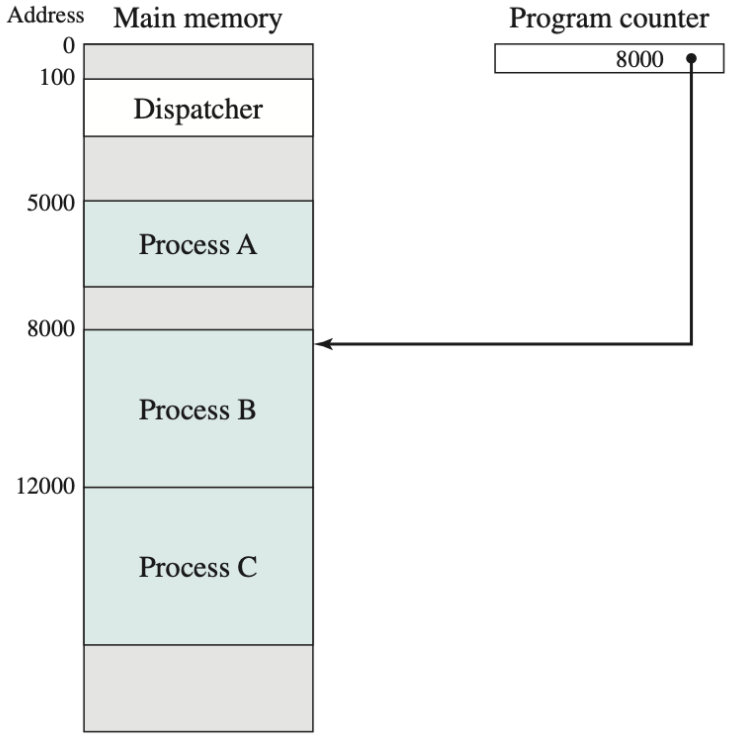




Multiproceso: ¿Virtualización?

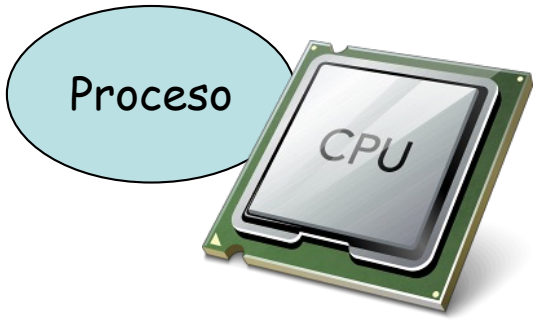
- Cuando varios procesos se ejecutan pero no disponemos de varias CPUs
- Cada proceso cree que dispone de la CPU pero se va alternando la ejecución entre procesos
- De nuevo se le está haciendo creer a alguien que dispone de ciertos recursos de forma exclusiva cuando no es así
- El Kernel del sistema operativo se encarga de gestionar el uso de los recursos físicos
- Gestiona pues el uso de la CPU y planifica la alternancia de los procesos en ejecución en la CPU

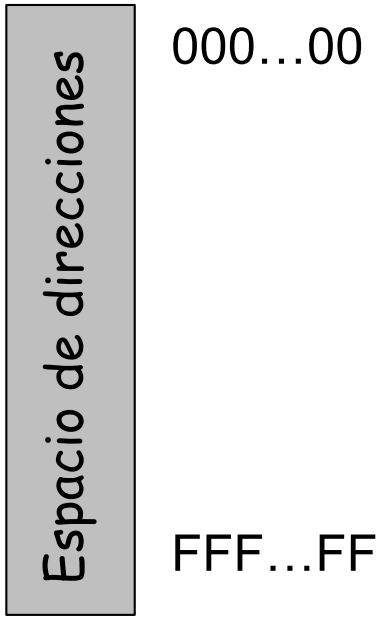




Virtualización de memoria

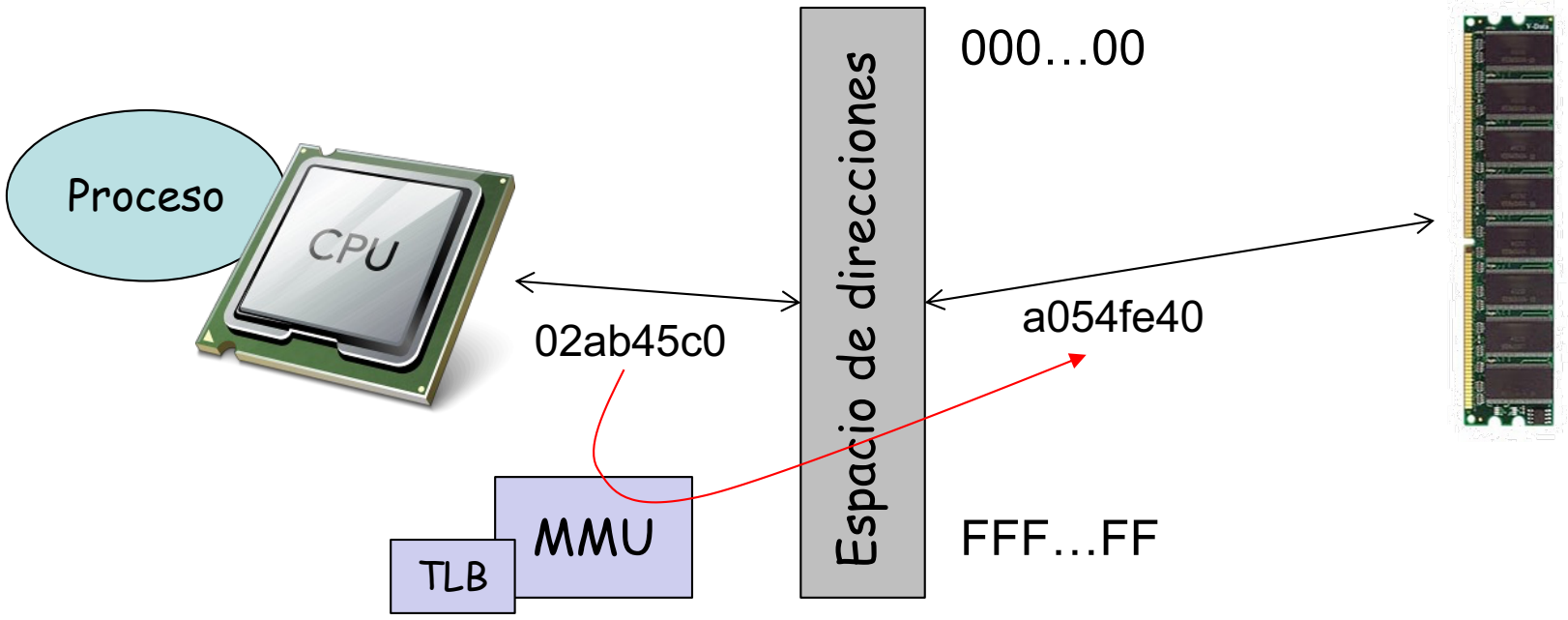
- El Kernel gestiona también el uso del recurso físico RAM
- Se puede conseguir que un proceso crea que dispone de toda la memoria
- De hecho podría ver más memoria que la existente
- Ve un espacio continuo de direcciones
- No puede acceder a la memoria en uso por otros procesos
- (…)





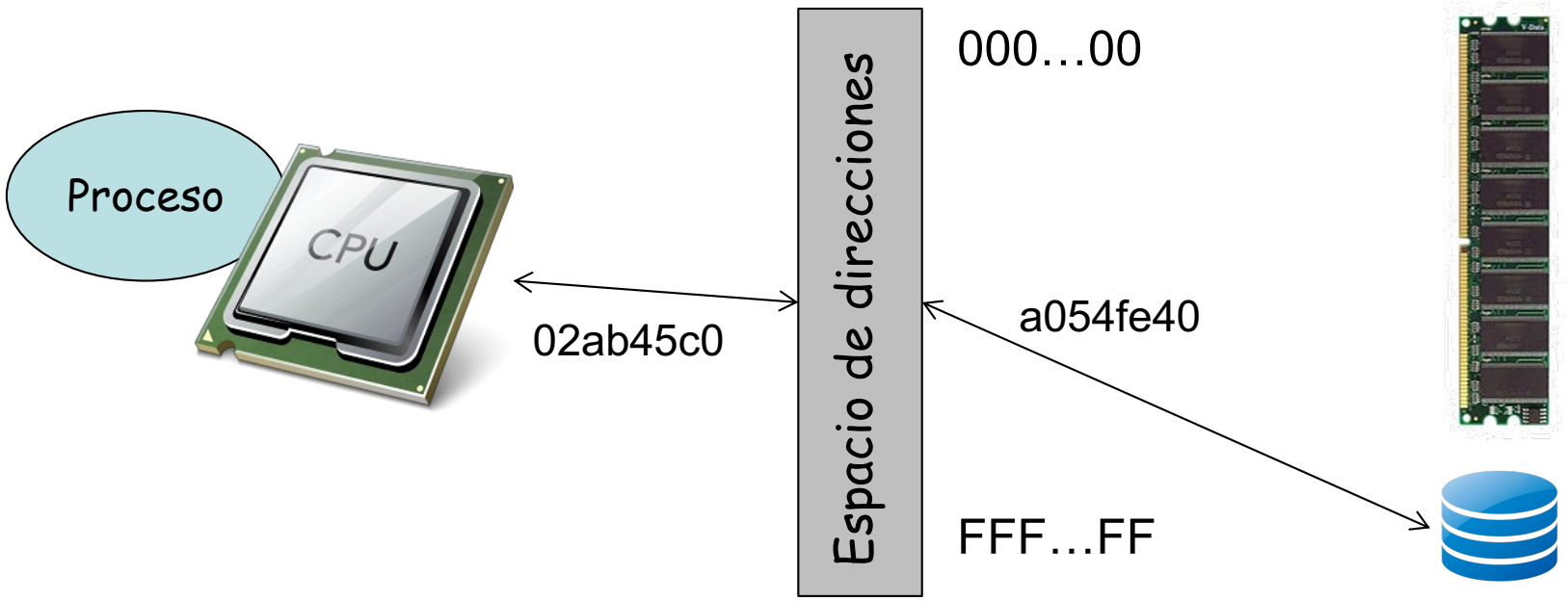
Virtualización de memoria

- Cuando la CPU intenta acceder a una dirección de memoria se debe convertir la dirección *virtual* en la dirección física
- Con esa dirección física se puede acceder a la RAM (ignorando las posibles caches)
- Esto se hace por “páginas” (hoy suelen ser de 4 KiB)
- Esta conversión la hace la MMU (*Memory Management Unit*)
- Hoy en día es parte de la CPU
- Es decir, necesitamos (o al menos mejora el rendimiento) apoyo del hardware



Virtualización de memoria

- El mapeo podría no llevar a memoria RAM sino a datos guardados en disco
- El disco es un dispositivo mucho más lento así que lo normal es mover los datos frecuentemente utilizados a RAM y los poco utilizados a disco

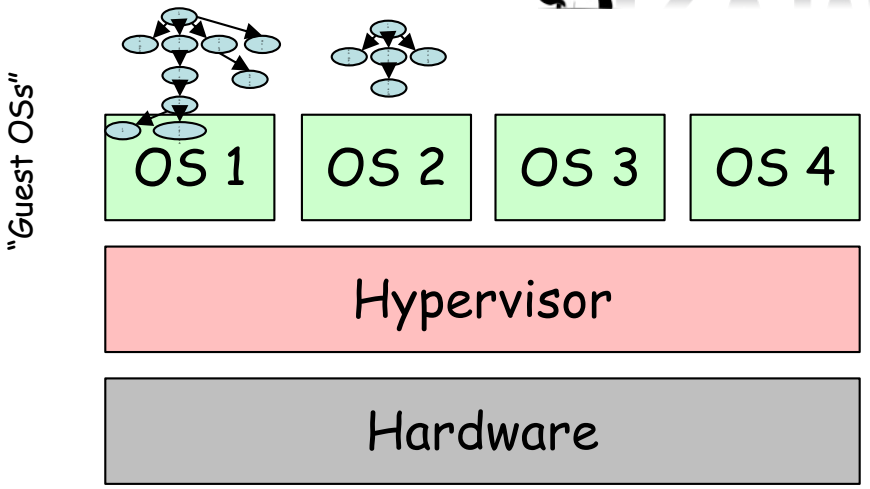


upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

El hypervisor

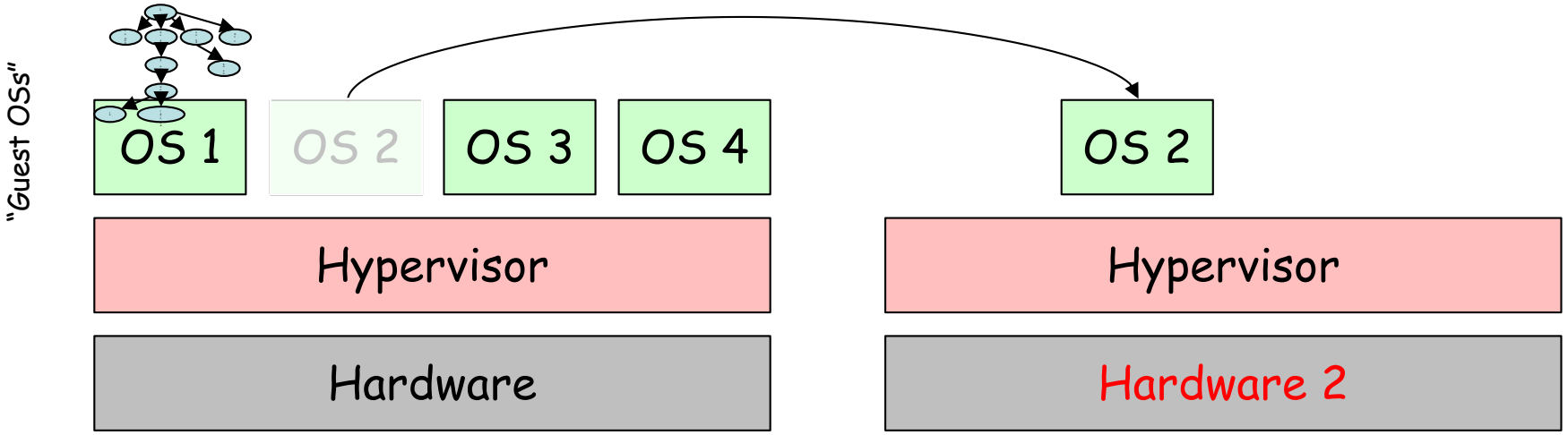
Hypervisor

- Es una capa software entre el hardware y el sistema operativo “guest”
- También llamado “*Virtual Machine Monitor*” (VMM)
- Oculta el hardware real y puede presentar diferente hardware a cada máquina virtual
- Esas máquinas virtuales no necesitan cambios para funcionar en otro hypervisor aunque emplee un hardware diferente siempre que les presente el mismo hardware virtual (…)



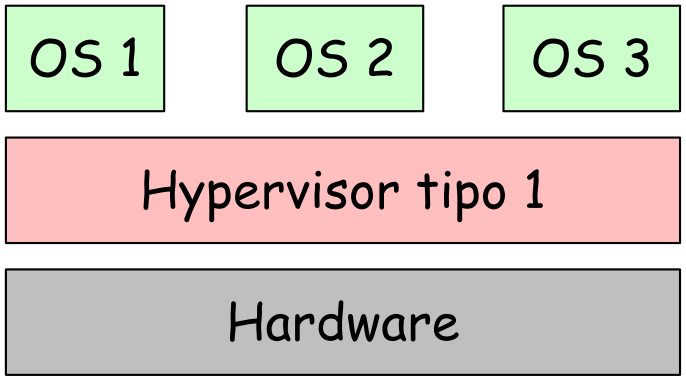
Hypervisor

- Es una capa software entre el hardware y el sistema operativo “guest”
- También llamado “*Virtual Machine Monitor*” (VMM)
- Oculta el hardware real y puede presentar diferente hardware a cada máquina virtual
- Esas máquinas virtuales no necesitan cambios para funcionar en otro hypervisor aunque emplee un hardware diferente siempre que les presente el mismo hardware virtual
- La máquina virtual, todo su sistema operativo instalado y las aplicaciones, puede ser un solo fichero, sencillo de copiar a otra máquina
- Esto lo haremos normalmente con la VM apagada



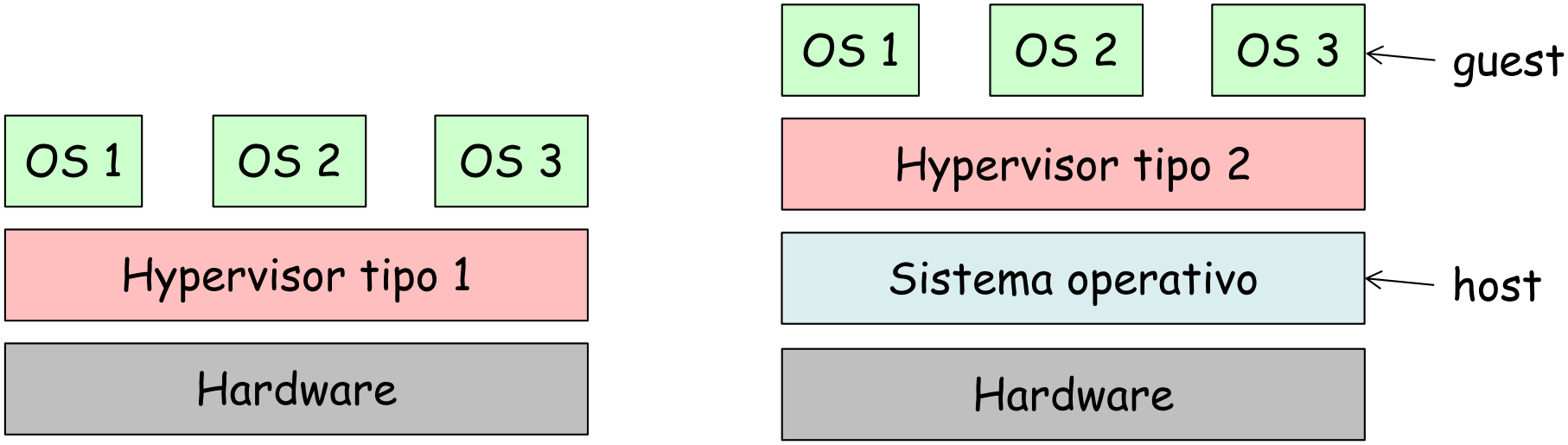
Tipos de Hypervisores

- Tipo 1, nativo o "*bare-metal*"
  - Se ejecuta directamente sobre el hardware
  - Controla dicho hardware
  - Consume poco espacio y memoria
  - El mejor rendimiento potencial
  - El hypervisor debe contar con drivers para el hardware
  - Ejemplos: Citrix XenServer, Vmware ESXi, Microsoft Hyper-V, Linux KVM
- (…)



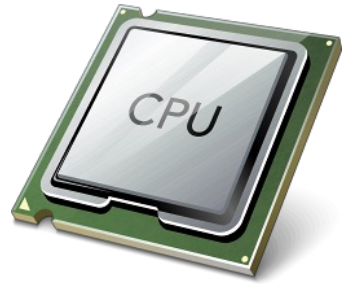
Tipos de Hypervisores

- Tipo 2 o “hosted”
  - El hypervisor corre como una aplicación sobre un sistema operativo convencional
  - El sistema operativo guest sobre el hypervisor
  - El sistema operativo host tiene un impacto en el rendimiento
  - Es más frecuente la existencia de drivers para el hardware
  - Ejemplos: VMware Workstation, VMware Server, Microsoft Virtual PC, Parallels Workstation, VirtualBox, QEMU



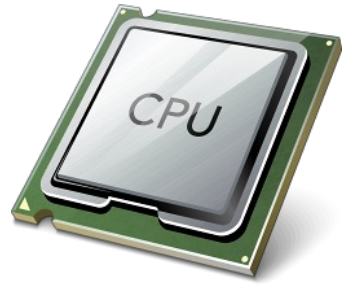
Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
  - Hace traducción (*on-the-fly*) de instrucciones (*binary translation*)
  - Se sustituyen las instrucciones no virtualizables por otras equivalentes
  - No requiere modificar el OS instalado
  - Ejemplos: VMware, Microsoft Virtual Server, Linux KVM, Parallels, VirtualBox, QEMU
- (…)



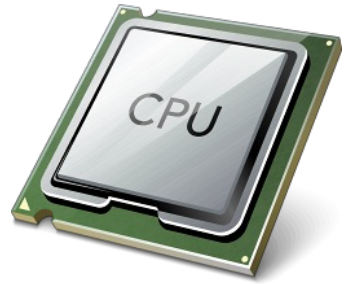
Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
- *Paravirtualization (OS assisted virtualization)*
  - Se modifica el sistema operativo guest sustituyendo las instrucciones no virtualizables
  - Requiere menos sobrecarga en ejecución pero hay que poder modificar el código de ese sistema operativo guest
  - Ejemplos: Xen, VMware (VMTools), Virtualbox (additions), UML
- *(…)*



Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
- *Paravirtualization (OS assisted virtualization)*
- *Hardware-assisted virtualization*
  - El hardware se encarga de la traducción de instrucciones privilegiadas
  - Requiere soporte por el hardware (Intel VT-x, AMD-V)
  - Ejemplos: VMware, Microsoft, Parallels, Xen, Virtualbox



Virtualización de RAM (doble)

- El sistema operativo guest emplea memoria virtual y la mapea a lo que él cree que es memoria física
- Eso no puede ser la auténtica memoria física, así que debe ser de nuevo mapeada
- *Shadow page tables* para hacerlo por soft o *nested paging* (Second Level Address Translation) por hardware si lo soporta la CPU
- Hay que virtualizar la MMU