

## Temas tratados al inicio de la clase

### Aspectos administrativos

- El primer examen quedará para semana 9-10.
- La tarea #1 será entregada por el profesor la semana siguiente.
- La tarea #1 será en grupos.
- Se debe traer la computadora a clase semana siguiente, el día exacto, se definirá en la siguiente lección.

### Repaso de los retos de las máquinas virtuales

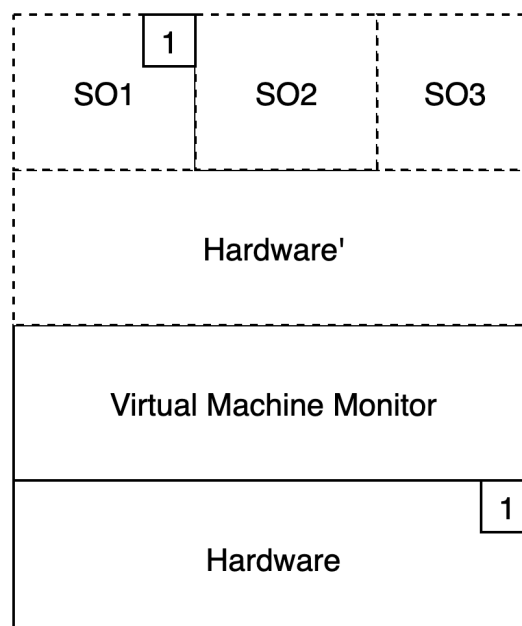
- Aislamiento.
- Memoria.
- CPU.
- Overhead.

IBM fue pionero en la creación de máquinas virtuales.

### Enfoques de uso de máquinas virtuales

El sistema operativo que es utilizado en los servidores de Azure para las máquinas virtuales, no es el mismo que una máquina virtual corriendo en una máquina local.

El esquema utilizado por Azure para sus máquinas virtuales es el siguiente

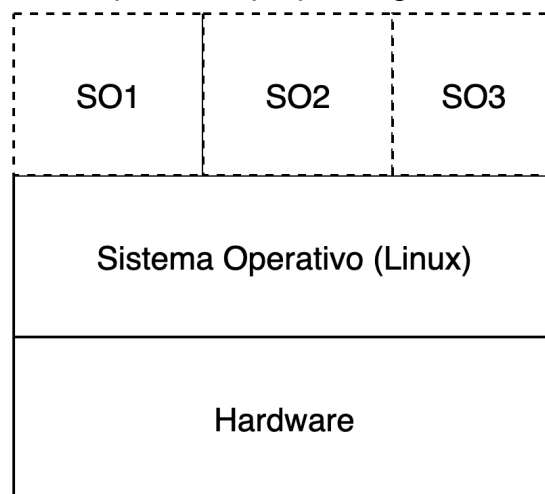


El Virtual Machine Monitor (VMM) es una capa de abstracción del hardware, no es la misma que la de una máquina de propósito general. Genera un hardware virtual para SO1, SO2 y SO3. Mediante esta abstracción Hardware' puede portear SO1,

SO2 y SO3. Si dichos sistemas operativos son muy diferentes, el porteo resulta más difícil para el VMM. Aquí se encuentra el HyperVisor.

Si se produce una interrupción en alguno de los sistemas operativos virtualizados, el VMM se encarga de dicha interrupción, sin embargo este “no sabe” qué hacer con eso por lo que consulta al sistema operativo, el cual generó la interrupción, por su tabla de interrupciones. Cuando la tiene, se verifica que el bit de validez esté activado tanto en el hardware físico, como en el sistema operativo que generó la interrupción. Una vez verificado, se procede a ejecutar la interrupción.

El enfoque utilizado por la máquinas de propósito general es el siguiente:



En este enfoque, el sistema operativo de uso general, ve a la máquina virtual como otro proceso, lo cual implica que se necesita calendarización para ejecutar las solicitudes de la máquina virtual.

Un ejemplo de esto, es el Taller #1 del curso, la razón por la que el programa implementado fué más rápido en la máquina virtual que en la local es porque máquina virtual es solamente se ejecuta el sistema operativo y el programa sin embargo, en la máquina local se pueden tener muchos procesos ejecutándose a la vez lo que retrasa la ejecución del programa.

## Se pasa a materia

### Procesos e Hilos en un Sistema Operativo

¿Se pueden ejecutar varias aplicaciones en la arquitectura básica del curso?

R/ Sí se puede, se realiza mediante calendarización (multiplexación de tiempo).

¿Cómo se comunican dos procesos?

R/ Mediante dos mecanismos:

- Memoria compartida, en este caso, el buffer de comunicación se encuentra en memoria.
- Intercambio de mensajes, en este caso, el buffer de comunicación se encuentra en el procesador.

¿Cómo se asignan los recursos?

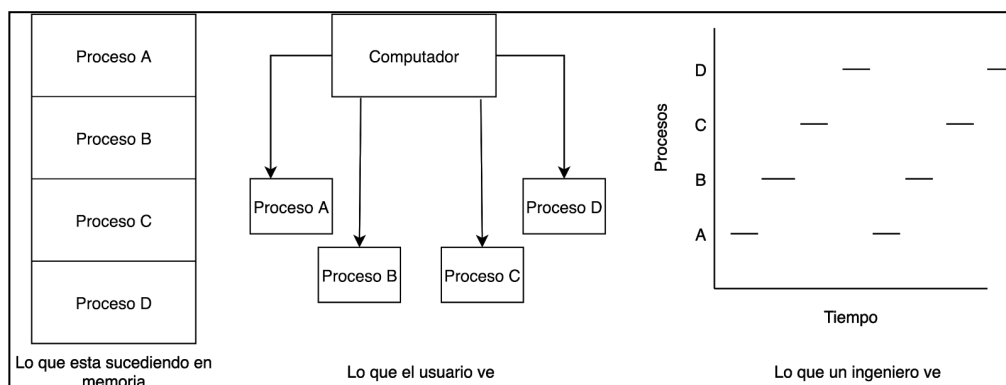
R/ El sistema operativo se encarga de administrar y asignar recursos a los procesos. Este tiene una tabla para llevar un control de que recurso está asignado a qué proceso. Esto últimos se identifican mediante el PID (identificador de procesos).

¿Qué pasa cuando un web server tiene muchos clientes a la vez y dan click al mismo tiempo?

R/ Se atienden y se crea un proceso para cada uno, esto sucede casi al instante.

### Procesos en un Sistema Operativo

- Es un concepto muy importante en el ámbito de los sistemas operativos.
- Es una abstracción de un programa en ejecución.
- Un programa no puede ejecutarse sin un proceso.
- Proporcionan la capacidad de operar concurrentemente en un único procesador.
- Todo software ejecutable en la computadora, se organiza en varios procesos (aplica más que todo a programas grandes).
- Formalmente, es una instancia de un programa, incluyendo los valores del contador de programa, los registros y las variables.
- Cada proceso posee un CPU virtual. Esto se refiere a que se debe asegurar que cada proceso corre como si estuviera solo en la memoria. A esto se le llama la garantía de la multiprogramación.



Se da la conmutación de procesos en el CPU (quitar uno y poner otro), a esto se le llama cambio de contexto.

¿Cuánto tarda el procesador en conmutar un proceso?

R/ No se puede saber, se necesitan más datos.

¿Qué se puede decir de la rapidez de los procesos en un procesador específico?

Es imposible reproducir la ejecución de un programa, debido a variables como la caché.

### **Proceso vs Programa**

Se tiene el caso de un cocinero científico que está horneando un pastel. Este científico tiene la receta y los instrumentos de cocina necesarios además, posee todos los ingredientes necesarios.

¿Quién es quién?

- La receta es el programa.
- El científico es el CPU.
- Los ingredientes son los datos de entrada.
- La cocina es el hardware necesario.
- El proceso es lo que lo hace posible.

### **Eventos**

El hijo del científico entra llorando, se cortó un dedo. El científico cuenta con un kit de primeros auxilios ¿qué debería hacer?.

R/ Hace un cambio de contexto, recordando que estaba haciendo.

Este caso ejemplifica una interrupción, esta se atiende y luego se regresa al flujo normal de operación.

### **Diferencia puntual entre proceso y programa**

El programa son las instrucciones (algoritmo), mientras que el proceso es la ejecución con los elementos necesarios (datos de entrada, hardware, contador, etc...).

El sistema operativo es un proceso.

### **¿Cuándo se crea un proceso en un Sistema Operativo?**

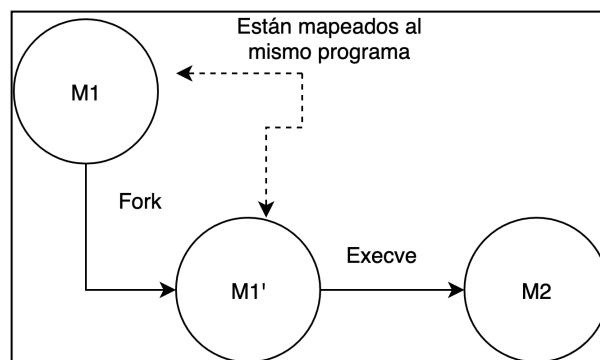
- En el boot.
- La ejecución, desde un proceso, de una llamada al sistema para la creación de otro proceso.
- Una petición de usuario para crear un proceso.
- El inicio de un trabajo por lotes.

## Creación de procesos.

Hay que hacer que un proceso ejecute una llamada al sistema. Esta llamada al sistema le indica al sistema operativo que cree un proceso y le indica a cuál programa asociarlo.

### Fork

- Es la llamada al sistema para crear un proceso en Unix.
- Crea un clon exacto del proceso que realizó la llamada.
- Los dos tienen la misma imagen de memoria.
- El proceso creado ejecuta otra *syscall* (*execve*) que cambia la imagen de memoria del proceso creado y ejecuta el programa correspondiente.



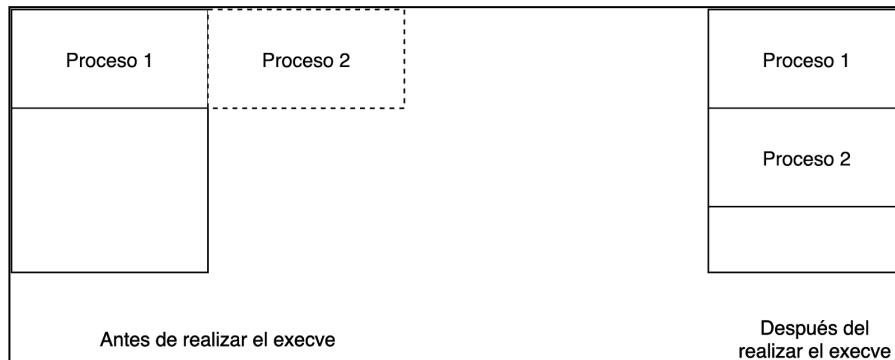
### CreateProcess

- Es la *syscall* de Windows para crear otro proceso.
- Crea un clon exacto del proceso que realizó la llamada y de una vez carga el programa correcto.
- Tiene más parámetros que fork, alrededor de 10.

Fork hace el paso intermedio para que tenga el mismo acceso y los mismos datos que el padre, para no tener que ir a disco a cargarlos, sino a memoria.

### CreateProcess vs Fork

- Los espacios de direccionamiento son diferentes.
- En Windows el espacio es distinto desde el inicio, en Unix se da el paso intermedio.
- No se comparte memoria, a excepción del inicio en Unix.



Una vez realizado el execve, el proceso padre no puede acceder a la memoria del proceso hijo y viceversa.

### Demonios

- Son procesos que permanecen en segundo plano para administrar tareas que el usuario no necesita saber.
- Generalmente, el sistema operativo posee docenas de demonios realizando tareas que son necesarias, pero que el usuario no sabe que se están ejecutando.
- Están relacionadas a los servicios.

### Terminación de procesos

Un proceso termina cuando ocurre:

- Una salida normal.
- Una salida por error.
- Un error fatal.
- Eliminado por otro proceso.

### Salida normal

El proceso termina porque ya terminó su trabajo. Un ejemplo es un compilador, cuando este termina su trabajo, realiza la `syscall` de `exit` en Unix o `ExitProcess` en Windows.

### Por error de usuario

El proceso termina porque se descubre un error. Un ejemplo es cuando el usuario ingresa de manera errónea los parámetros para compilar un archivo en C, en este caso se produce el `syscall` para terminar el proceso.

### Error fatal

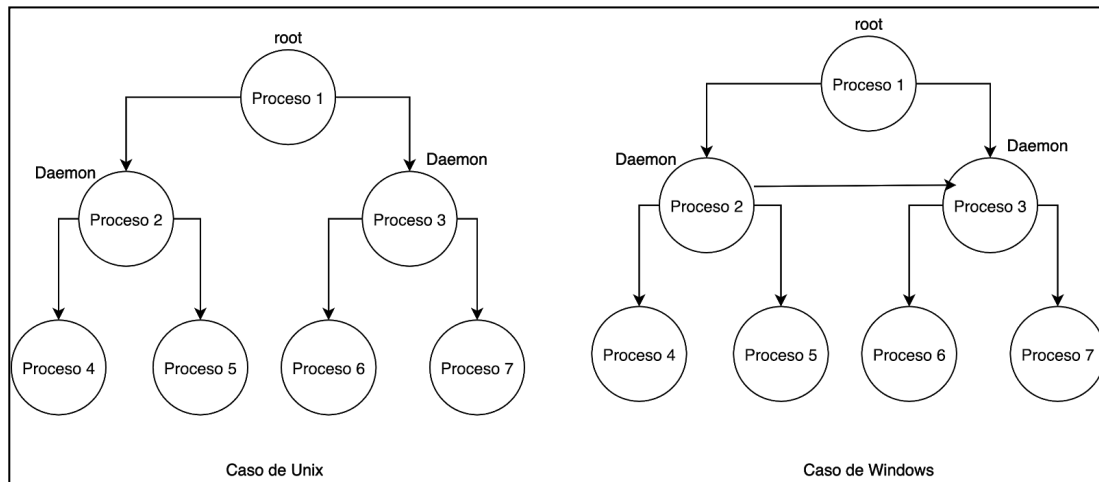
Se refiere a un error en el programa (interrupción sincrónica). Un ejemplo es cuando ocurre una división por cero.

## Eliminado por otro proceso

Cuando un proceso ejecuta una *syscall* para eliminar otro proceso. En el caso de Unix es *kill* y en Windows es *TerminateProcess*.

## Jerarquía en los procesos

¿Qué pasa cuando se crea un proceso en Linux?



Se dice que Windows no tiene jerarquía de procesos, porque permite cambios de referencia, ya que, en el caso de la imagen, el proceso 2 puede referenciar al proceso 3 mediante una llamada, sin que exista una relación padre-hijo entre ellos.