

# Sistemas operativos

Pedro O. Pérez M., PhD.

Diseño de sistemas embebidos avanzados  
Tecnológico de Monterrey

*pperezm@tec.mx*

09-2022

## ① Introducción

Operación de un sistema computacional

Arquitectura de un sistema computacional

## ② Estructura de un sistema operativo

Multiprogramación

Procesos

## ③ Caché

Introducción

Coherencia de caché

## ④ Ambientes Computacionales

Cómputo Tradicional-Móvil-Distribuido

Cómputo Cliente-Servidor, Punto-a-Punto

Virtualización, Cómputo en la Nube

Sistemas Embebidos en Tiempo Real

# ¿Qué es un sistema operativo?

- Un sistema operativo es un programa que administra el hardware de una computadora. También proporciona una base para los programas de aplicación y actúa como intermediario entre el usuario de la computadora y el hardware de la computadora.
- Un aspecto sorprendente de los sistemas operativos es la forma en que varían para realizar estas tareas:
  - Mainframes: Optimizar la utilización del hardware.
  - PC. Ejecutar juegos complejos, aplicaciones comerciales, etc.
  - Móviles: Proporcionar un entorno en el que usuario pueda interactuar fácilmente con la computadora para ejecutar programas.
- Por lo tanto, algunos sistemas operativos están diseñados para ser convenientes, otros para ser eficientes y otros para ser una combinación de los dos.

# ¿Qué hace un sistema operativo?

El sistema operativo controla el hardware y coordina su uso entre los diversos programas de aplicación para los distintos usuarios. El sistema operativo proporciona los medios para el uso adecuado de estos recursos en el funcionamiento del sistema computacional.

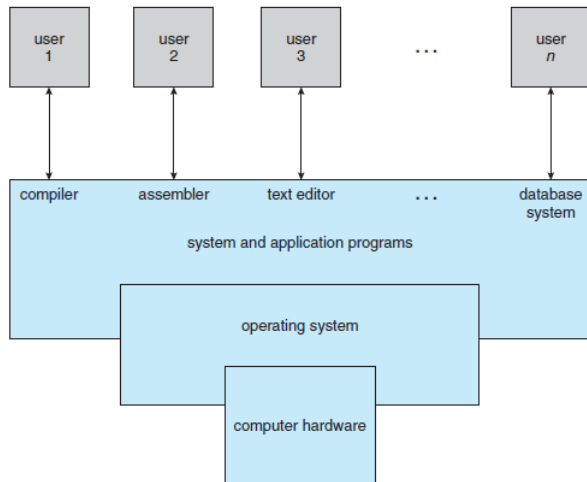


Figure 1.1 Abstract view of the components of a computer system.

# Operación de un sistema computacional

- Cuando una computadora se enciende, inmediatamente se ejecuta un programa inicial. Este programa inicial (bootstrap) tiende a ser simple y, por lo general, se almacena dentro de hardware es una memoria de solo lectura (ROM, EEPROM), conocida por el termino general de firmware.
  - Primero, se encarga de inicializar todos los aspectos de la computadora.
  - Segundo, debe ubicar el núcleo del sistema operativo (kernel) y cargarlo en memoria.

- Una vez que el kernel se carga y se ejecuta, puede comenzar a brindar servicios al sistema y a sus usuarios. Algunos servicios se proporcionan fuera del kernel, mediante programas del sistema que se cargan en la memoria en el momento del arranque para convertirse en procesos del sistema, o demonios del sistema que se ejecutan todo el tiempo que se ejecuta el kernel.
- La ocurrencia de un evento generalmente se indica mediante una interrupción del hardware o del software. El hardware puede desencadenar una interrupción en cualquier momento enviando una señal al CPU, generalmente a través del bus del sistema. El software puede desencadenar una interrupción al ejecutar una operación especial llamada llamada al sistema.

- Cuando se interrumpe al CPU, detiene lo que está haciendo y transfiere inmediatamente la ejecución a una ubicación fija. La ubicación fija generalmente contiene la dirección de inicio donde se encuentra la rutina de servicio para la interrupción. Se ejecuta la rutina del servicio de interrupción; al finalizar, el CPU reanuda el cálculo interrumpido.





# Arquitectura de un sistema computacional

- Hasta hace poco, la mayoría de los sistemas informáticos utilizaban un solo procesador. En un sistema de un solo procesador, hay un CPU principal capaz de ejecutar un conjunto de instrucciones de propósito general, incluidas las instrucciones de los procesos del usuario.
- En los últimos años, los sistemas multiprocesador (también conocidos como sistemas paralelos o sistemas multinúcleo) han comenzado a dominar el panorama de la informática. Dichos sistemas tienen dos o más procesadores en comunicación cercana, compartiendo el bus de la computadora y, a veces, el reloj, la memoria y los dispositivos periféricos.

Los sistemas multiprocesador tienen tres ventajas principales:

- ① Mayor rendimiento. Al aumentar la cantidad de procesadores, esperamos hacer más trabajo en menos tiempo.
- ② Economía de escala. Los sistemas multiprocesador pueden costar menos que los sistemas equivalentes de un solo procesador, ya que pueden compartir periféricos, almacenamiento masivo y fuentes de alimentación
- ③ Mayor confiabilidad. Si las funciones se pueden distribuir correctamente entre varios procesadores, entonces la falla de un procesador no detendrá el sistema, solo lo ralentizará.

Los sistemas de procesadores múltiples que se utilizan hoy en día son de dos tipos:

- ① Algunos sistemas utilizan multiprocesamiento asimétrico, en el que a cada procesador se le asigna una tarea específica. El procesador principal controla el sistema; los otros procesadores miran al jefe en busca de instrucciones o tienen tareas predefinidas.
- ② Los sistemas más comunes utilizan multiprocesamiento simétrico (SMP), en el que cada procesador realiza todas las tareas dentro del sistema operativo. SMP significa que todos los procesadores son iguales; no existe una relación jefe-trabajador entre los procesadores.

Otro tipo de sistema multiprocesador es un sistema en clúster, que reúne varios CPU. Los sistemas agrupados se diferencian de los sistemas multiprocesador en que están compuestos por dos o más sistemas individuales, o nodos, unidos. Estos sistemas se consideran débilmente acoplados. Cada nodo puede ser un sistema de un solo procesador o un sistema multinúcleo.

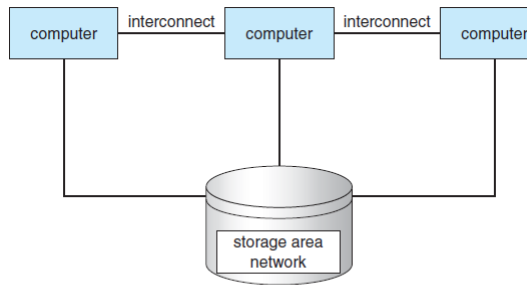


Figure 1.8 General structure of a clustered system.

- Uno de los aspectos más importantes de los sistemas operativos es la capacidad de multiprogramación. En general, un solo programa no puede mantener ocupados ni al CPU ni a los dispositivos de E/S en todo momento. Los usuarios individuales suelen tener varios programas en ejecución. La multiprogramación aumenta la utilización de la CPU al organizar los trabajos (código y datos) para que la CPU siempre tenga uno para ejecutar.

La idea es la siguiente:

- El sistema operativo mantiene varios trabajos en la memoria simultáneamente. Dado que, en general, la memoria principal es demasiado pequeña para acomodar todos los trabajos, los trabajos se guardan inicialmente en el disco el “pool” de trabajos.

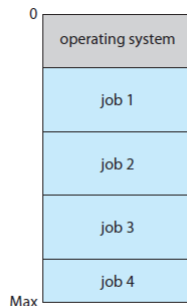


Figure 1.9 Memory layout for a multiprogramming system.

- El conjunto de trabajos en la memoria puede ser un subconjunto de los trabajos que se mantienen en el “pool” de trabajos. El sistema operativo selecciona y comienza a ejecutar uno de los trabajos en la memoria. Eventualmente, es posible que el trabajo tenga que esperar a que se complete alguna tarea, como una operación de E/S. En un sistema no multiprogramado, el CPU permanecería inactiva. En un sistema multiprogramado, el sistema operativo simplemente cambia y ejecuta otro trabajo. Cuando ese trabajo necesita esperar, la CPU cambia a otro trabajo y así sucesivamente. Finalmente, el primer trabajo termina de esperar y recupera la CPU. Mientras sea necesario ejecutar al menos un trabajo, la CPU nunca estará inactiva.

- Un programa no hace nada a menos que sus instrucciones sean ejecutadas por una CPU. Un programa en ejecución es un proceso.
- El proceso necesita ciertos recursos, incluido el tiempo de CPU, la memoria, los archivos y los dispositivos de E/S, para realizar su tarea. Estos recursos se le dan al proceso cuando se crea o se le asignan mientras se está ejecutando.
- Un proceso es la unidad de trabajo en un sistema. Un sistema consta de una colección de procesos, algunos de los cuales son procesos del sistema operativo (los que ejecutan el código del sistema) y el resto son procesos del usuario (los que ejecutan el código del usuario).



El sistema operativo es responsable de las siguientes actividades en relación con la gestión de procesos:

- Programar procesos e hilos en los CPUs.
- Crear y eliminar procesos de usuario y del sistema.
- Suspender y reanudar procesos.
- Proporcionar mecanismos para la sincronización de procesos.
- Proporcionar mecanismos para la comunicación de procesos.

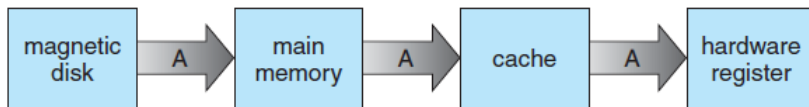
El almacenamiento en caché es un principio importante de los sistemas de cómputo. Así es como funciona. La información normalmente se guarda en algún sistema de almacenamiento (como la memoria principal). A medida que se utiliza, se copia a un sistema de almacenamiento más rápido, el caché, de forma temporal. Cuando necesitamos una información en particular, primero verificamos si está en la caché. Si es así, usamos la información directamente del caché. Si no es así, usamos la información de la fuente, poniendo una copia en la caché bajo el supuesto de que la necesitaremos nuevamente pronto.

Debido a que las memorias caché tienen un tamaño limitado, la administración de la memoria caché es un problema de diseño importante. La selección cuidadosa del tamaño de la caché y de una política de reemplazo puede resultar en un rendimiento mucho mayor.

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

**Figure 1.11** Performance of various levels of storage.

En una estructura de almacenamiento jerárquica, los mismos datos pueden aparecer en diferentes niveles del sistema de almacenamiento. Por ejemplo, suponga que un entero A que debe incrementarse en 1 se encuentra en el archivo B y el archivo B reside en un disco magnético. La operación de incremento procede emitiendo primero una operación de E/S para copiar el bloque de disco en el que A reside en la memoria principal. A esta operación le sigue la copia de A en la caché y en un registro interno. Una vez que se produce el incremento en el registro interno, el valor de A difiere en los distintos sistemas de almacenamiento. El valor de A se vuelve el mismo solo después de que el nuevo valor de A se escribe desde el registro interno de regreso al disco magnético.



**Figure 1.12** Migration of integer A from disk to register.

- En un entorno donde solo se ejecuta un proceso a la vez, esta disposición no plantea dificultades, ya que un acceso al entero A siempre será a la copia en el nivel más alto de la jerarquía.
- Sin embargo, en un entorno multitarea, donde el CPU se alterna entre varios procesos, se debe tener mucho cuidado para garantizar que, si varios procesos desean acceder a A, cada uno de estos procesos obtendrá el valor actualizado más recientemente de A .

La situación se complica en un entorno multiprocesador donde, además de mantener registros internos, cada una de las CPU también contiene una caché local. En tal entorno, una copia de A puede existir simultáneamente en varias cachés. Dado que los distintos CPU pueden ejecutarse todas en paralelo, debemos asegurarnos de que una actualización del valor de A en una caché se refleje inmediatamente en todas las demás cachés donde A reside. Esta situación se denomina **coherencia de caché** y suele ser un problema de hardware (que se maneja por debajo del nivel del sistema operativo).

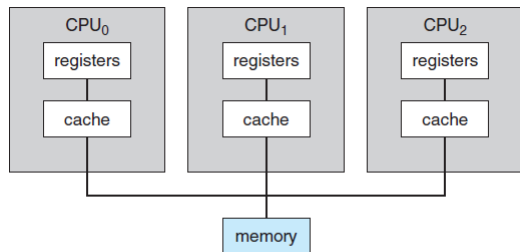


Figure 1.6 Symmetric multiprocessing architecture.

A medida que la computación ha madurado, las líneas que separan muchos de los entornos computacionales tradicionales se han difuminado. Considera el “entorno de oficina típico”. Hace apenas unos años, este entorno consistía en PC conectadas a una red, con servidores que proporcionaban servicios de archivos e impresión. El acceso remoto era incómodo y la portabilidad se logró mediante el uso de computadoras portátiles. Los terminales conectados a mainframes también prevalecían en muchas empresas, con incluso menos opciones de portabilidad y acceso remoto.



La informática móvil se refiere a la informática en teléfonos inteligentes de mano y tabletas. Estos dispositivos comparten las características físicas distintivas de ser portátiles y livianos.

- Un sistema distribuido es una colección de sistemas informáticos físicamente separados, posiblemente heterogéneos, que están conectados en red para proporcionar a los usuarios acceso a los diversos recursos que mantiene el sistema. El acceso a un recurso compartido aumenta la velocidad de cálculo, la funcionalidad, la disponibilidad de datos y la confiabilidad.
- Una red, en los términos más simples, es una ruta de comunicación entre dos o más sistemas. Los sistemas distribuidos dependen de las redes para su funcionalidad.
  - Local-area Network (LAN).
  - Wide-area Network (WAN).
  - Metropolitan-area Network (MAN).
  - Personal-area Network (PAN).

A medida que las PC se han vuelto más rápidas, más potentes y más baratas, los diseñadores se han alejado de la arquitectura del sistema centralizado. Los terminales conectados a sistemas centralizados ahora están siendo reemplazados por PC y dispositivos móviles. En consecuencia, la funcionalidad de la interfaz de usuario que alguna vez fue manejada directamente por sistemas centralizados está siendo manejada cada vez más por PC, con bastante frecuencia a través de una interfaz web. Como resultado, muchos de los sistemas actuales actúan como sistemas de servidor para satisfacer las solicitudes generadas por los sistemas de los clientes. Esta forma de sistema distribuido especializado, llamado sistema cliente-servidor.

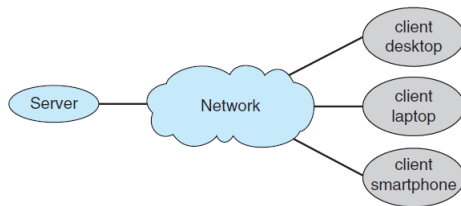
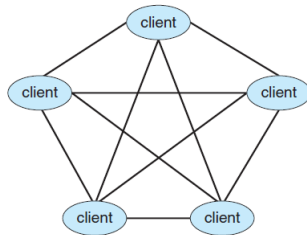


Figure 1.18 General structure of a client-server system.

Otra estructura para un sistema distribuido es el modelo de sistema peer-to-peer (P2P). En este modelo, los clientes y los servidores no se distinguen entre sí. En cambio, todos los nodos dentro del sistema se consideran pares y cada uno puede actuar como cliente o servidor, dependiendo de si está solicitando o brindando un servicio. Los sistemas peer-to-peer ofrecen una ventaja sobre los sistemas cliente-servidor tradicionales. En un sistema cliente-servidor, el servidor es un cuello de botella; pero en un sistema de igual a igual, los servicios pueden ser proporcionados por varios nodos distribuidos por la red.



**Figure 1.19** Peer-to-peer system with no centralized service.

La virtualización es una tecnología que permite que los sistemas operativos se ejecuten como aplicaciones dentro de otros sistemas operativos. A primera vista, parece haber pocas razones para tal funcionalidad. Pero la industria de la virtualización es vasta y está creciendo, lo que demuestra su utilidad e importancia.

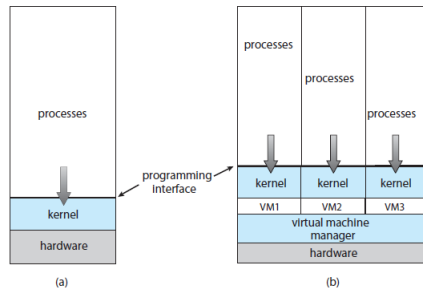


Figure 1.20 VMware.

La computación en la nube es un tipo de computación que ofrece computación, almacenamiento e incluso aplicaciones como un servicio a través de una red. De alguna manera, es una extensión lógica de la virtualización, porque utiliza la virtualización como base para su funcionalidad.

En realidad, existen muchos tipos de computación en la nube, incluidos los siguientes:

- Red privada.
- Red pública.
- Red híbrida.

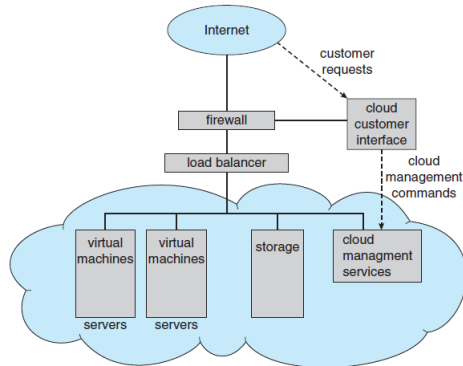


Figure 1.21 Cloud computing.

## Tipos de servicios en la nube:

- Software como servicio (SaaS): una o más aplicaciones (como procesadores de texto u hojas de cálculo) disponibles a través de Internet.
- Plataforma como servicio (PaaS): una pila de software lista para el uso de aplicaciones a través de Internet (por ejemplo, un servidor de base de datos).
- Infraestructura como servicio (IaaS): servidores o almacenamiento disponibles a través de Internet (por ejemplo, almacenamiento disponible para realizar copias de seguridad de los datos de producción).

Las computadoras embebidas son la forma más común de computadoras que existe. Estos dispositivos se encuentran en todas partes, desde motores de automóviles y robots de fabricación hasta DVD y hornos microondas. Suelen tener tareas muy específicas. Los sistemas en los que se ejecutan suelen ser primitivos, por lo que los sistemas operativos proporcionan funciones limitadas. Por lo general, tienen poca o ninguna interfaz de usuario, y prefieren dedicar su tiempo a monitorear y administrar dispositivos de hardware, como motores de automóviles y brazos robóticos.



- Estos sistemas embebidos varían considerablemente. Algunos son computadoras de propósito general que ejecutan sistemas operativos estándar, como Linux, con aplicaciones de propósito especial para implementar la funcionalidad. Otros son dispositivos de hardware con un sistema operativo integrado de propósito especial que proporciona solo la funcionalidad deseada.
- Los sistemas embebidos casi siempre ejecutan sistemas operativos en tiempo real. Se utiliza un sistema en tiempo real cuando se han impuesto requisitos de tiempo rígidos en la operación de un procesador o el flujo de datos; por tanto, a menudo se utiliza como dispositivo de control en una aplicación dedicada.

En grupos de 2 integrantes,

- Investiguen cuál el sistema operativo más utilizado en sistemas embebidos.
- Seleccionen uno, investiguen sus características y qué sistemas se utilizan.
- Generen un reporte con los resultados, suban el reporte a Canvas.