

# 1. Introducción a los Sistemas Operativos

## 1. Introducción

Un Sistema operativo es un programa que administra el hardware de una computadora. También proporciona las bases para los programas de aplicación, y actúa como intermediario entre el usuario y el hardware. Estas tareas, pueden ser llevadas a cabo de varias formas, lo que permite que algunos sistemas Operativos se diseñen para ser prácticos, otros eficientes y otros para ser ambas cosas.

Dado que un sistema operativo es un software grande y complejo, debe crearse pieza por pieza. En este curso veremos, entre otras cosas, una introducción a los principales componentes de un sistema operativo.

## 2. ¿Qué hace un sistema Operativo?

Un sistema informático puede dividirse en cuatro componentes: el hardware, el sistema Operativo, los programas de aplicación, y los usuarios. El Sistema operativo controla y coordina el uso del hardware entre los diversos programas de aplicación por parte de los distintos usuarios.

También podemos ver un sistema informático como hardware, software y datos. El sistema operativo proporciona los medios para hacer un uso adecuado de estos recursos durante el funcionamiento del sistema informático.

## 3. Definición Sistema Operativo

Un sistema operativo es un programa, o conjunto de programas que torna amigable, eficiente y productivo el uso de un computador (hardware), permitiendo la ejecución aplicaciones de usuario. Es el intermediario entre las aplicaciones de usuario y el hardware.

Metas:

- Brindar un ambiente de realización y ejecución de aplicaciones
- Proveer un entorno sin interferencias a cada usuario (interferencia: Lo que un usuario modifica en su entorno, no interfiera ni modifique lo de otro usuario)
- Administrar en forma equitativa los recursos(hardware y software)
- Hacerlo de la forma mas amigable e intuitiva posible

Todas las aplicaciones de usuario requieren un conjunto común de operaciones que son incorporadas al sistema operativo.

Tareas principales:

- Implementar diferentes entornos para diferentes usos (interfaz grafica, shells, tipo web, etc)
- Proveer una o más interfaces con el usuario.
- Proveer a las aplicaciones un conjunto de servicios(a través de los “system services”)
- Eficiencia y equidad en la administración de recursos.

Se Puede decir que el Sistema operativo es un:

- Administrador de recursos. Sus tareas consiste en administrar los recursos disponibles y decidir como asignar estos recursos según los pedidos y asignaciones que tenga.
- Programa de Control: Controla la ejecución de los programas para la prevención de errores y mal uso del sistema.

Frecuentemente la porción residente (el código incambiable. Cuanto más código incambiable, más inflexible es) del propio sistema operativo se denomina *núcleo del sistema* (Kernel).

#### **4. Evolución histórica de los Sistemas Operativos**

##### SISTEMAS BATCH Ó POR LOTES (70' y comienzos de los 80'):

En las primeras épocas los sistemas eran grandes y costosos. Constaban de una entrada de trabajos y una salida impresa, por lo cual la interacción con el usuario era prácticamente nula. Las principales características eran que el sistema soportaba de un único trabajo a la vez, y que las tareas relacionadas se agrupaban en conjuntos o lotes, para su procesamiento más eficiente.

A comienzos de los 80', utilizando las técnicas de Spooling (proceso mediante el cual la computadora introduce trabajos en un buffer, de manera que un dispositivo pueda acceder a ellos cuando esté listo) y multiprogramación (ejecución de múltiples tareas compartiendo recursos) se pudo comenzar a desarrollar técnicas de planificación de despacho.

Esta técnica consistía en seleccionar un lote de trabajos que estaban en memoria secundaria para cargarlos en memoria principal. Luego, el SO seleccionaba uno de ellos para ejecutar, y si este debía esperar por alguna tarea (por ejemplo ejecución de E/S) el sistema elegía otro del lote para utilizar el procesador. Esto, incrementó el uso del procesador.

##### SISTEMAS DE TIEMPO COMPARTIDO (Fines 80'):

Estos Sistemas eran multiusuarios. Ejecutaban programas de forma concurrente con una elevada tasa de despacho de procesador (cambio de contexto), de forma tal de permitir a los usuarios que interactúen directamente con el sistema como si fueran su único usuario.

La necesidad de acceder y actualizar datos de forma concurrente, creó la necesidad de evolucionar el sistema de archivos a uno multiusuario, incorporando técnicas de protección de accesos.

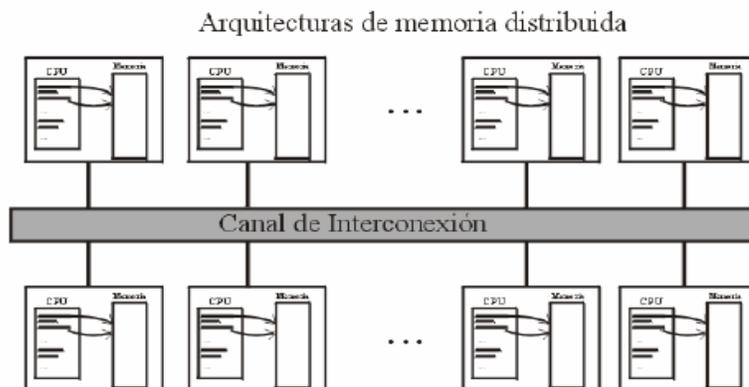
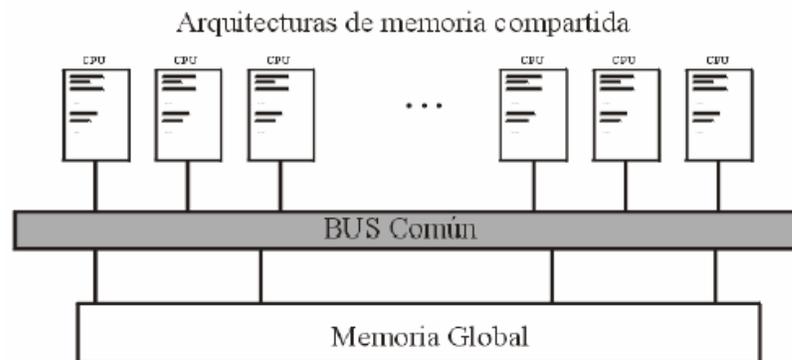
##### SISTEMAS PARA COMPUTADORES PERSONALES (80'):

Con costos de hardware decrecientes, fue posible el diseño y uso de computadores personales. Los Sistemas fueron diseñados en base a que serían utilizados por un único usuario, y todo el énfasis en el desarrollo estuvo en mejorar la interacción con el usuario. Se desarrolló la interfaz de ventanas que conocemos hoy.

## SISTEMAS PARALELOS (comienzos 90'):

Son Sistemas donde se dispone de más de un procesador, permitiendo ejecución simultánea y sincronizada de procesos. Se Clasifican en:

- Altamente integrados: “tightly coupled”. Son sistemas en donde los canales de interconexión son de alta velocidad. (bus común o memoria compartida)
- Poco Integrados: “closely coupled”. Son sistemas en donde los canales de interconexión son de baja velocidad (sistemas en red)



Veamos ahora otra clasificación de los Sistemas paralelos:

- Asimétricos: Se designa un CPU (master) para ejecutar el código del núcleo, para no lidiar con la concurrencia, los demás (slaves) ejecutarán lo que éste les designe.
- Simétricos: Todos los procesadores son considerados iguales, el código del núcleo se dispone en memoria común y es ejecutado por cualquier procesador.

Y otra clasificación más:

- UMA (Uniform Memory Access): Cada CPU accede a cualquier lugar de la memoria en el mismo tiempo.
- NUMA (Non-Uniform Memory Access): Los CPU tienen áreas de memoria a las que acceden más rápido que el resto.

Veamos ahora una clasificación de Arquitecturas (Taxonomía de Flynn):

- SISD (Single Instruction, Single Data): Arquitectura secuencial, por ende no hay paralelismo; son arquitecturas monoprocesadores.
- SIMD (Single Instruction, Multiple Data): Son sistemas que ejecutan la misma instrucción sobre un conjunto de datos (Arquitectura vectorial).
- MISD (Multiple Instruction, Single Data): Paralelismo redundante.
- MIMD (Multiple Instruction, Multiple Data): Varios procesadores autónomos que ejecutan en forma simultánea varias instrucciones sobre datos diferentes (puede ser memoria compartida: escalan poco, acceso a memoria es cuello de botella; memoria distribuida: escalan a miles de procesadores, conectados en una red de alta velocidad)

Como ejemplo de sistemas computacionales que utilizan sistemas paralelos tenemos los clusters. Estos son sistemas en los que participan varias computadoras. Los Clusters brindan alta disponibilidad (mantiene una serie de servicios, a pesar de posibles fallos), alto rendimiento (en cuanto a capacidad de cálculo) y balance de carga (técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, etc)

Se Clasifican en:

- Simétricos: Todos los nodos ejecutan tareas y asumen las de otros ante fallas.
- Asimétricos: Nodos primarios ejecutan tareas y nodos secundarios esperan fallas.

#### SISTEMAS DE TIEMPO REAL:

Son sistemas en los cuales todo resultado debe producirse en un cierto tiempo. De lo contrario se considera que el sistema ha fallado.