

en cierto nivel, pero internamente se rescribió desde cero. El otro competidor importante es **UNIX**, que domina en las estaciones de trabajo y otras computadoras del extremo alto, como los **servidores de red**. **UNIX** es popular sobre todo en máquinas basadas en microprocesadores **RISC** de alto rendimiento. Estas máquinas por lo regular tienen la potencia de cómputo de una mini computadora, a pesar de estar dedicadas a un solo usuario, por lo que resulta lógico que estén equipadas con un sistema operativo diseñado originalmente para mini computadoras, a saber, **UNIX**.

- **Redes de computadoras.**- una tendencia interesante que apareció a mediados de la década de **1980** fue el crecimiento de **redes de computadoras** personales en las que se ejecutan **sistemas operativos de red** o **sistemas operativos distribuidos**.

En un **sistema operativo de red** los usuarios están conscientes de la existencia de múltiples computadoras y pueden ingresar en máquinas remotas y copiar archivos de una máquina a otra. Cada máquina ejecuta su propio sistema operativo local y tiene su propio usuario o usuarios locales.

Los **sistemas operativos de red** no son fundamentalmente distintos de aquellos para un solo procesador. Obviamente, estos sistemas necesitan un controlador de la interfaz con la red y software de bajo nivel para operarlo, así como programas para realizar inicios de sesión remotos y acceso a archivos remotos, pero estas adiciones no alteran la estructura esencial del sistema operativo.

Un **sistema operativo distribuido**, en cambio, presenta el mismo aspecto a los usuarios que un sistema tradicional de un solo procesador, aunque en realidad se compone de múltiples procesadores. Los usuarios no deben enterarse de en dónde se están ejecutando sus programas o almacenando sus archivos; de todo eso debe encargarse el sistema operativo **automática y eficientemente**.

Los verdaderos **sistemas operativos distribuidos** requieren más que la adición de un poco más de código a un sistema operativo uniprocador, porque los sistemas distribuidos y centralizados difieren en aspectos cruciales. Los **sistemas distribuidos**, por ejemplo, a menudo permiten a las aplicaciones ejecutarse en varios procesadores al mismo tiempo, por lo que requieren algoritmos de planificación de procesador más complejos, a fin de optimizar el grado de **paralelismo**.

En muchos casos, los **retardos de comunicación** dentro de la red implican que estos (y otros) algoritmos deban ejecutarse con información incompleta, caduca o incluso incorrecta. Esta situación difiere radicalmente de un sistema de un solo procesador en el que el sistema operativo tiene toda la información sobre el estado del sistema.

1.3 Términos Básicos.

A continuación describimos algunos términos importantes que usaremos a lo largo del curso.

- **Sistemas por lotes.**- cuando se desarrollaron por primera vez, estaban caracterizados por la agrupación en bloques de trabajos similares. Los modernos sistemas utilizan otras características. El rasgo característico de un sistema por lotes es la ausencia de interacción entre el usuario y el trabajo mientras éste se ejecuta. El trabajo se prepara y se envía. Tiempo después aparece la salida.
- **Multiprogramación.**- un solo usuario no puede, en general, mantener todo el tiempo ocupado a la unidad central de procesamiento ó a los dispositivos de entrada/salida. La **multiprogramación** aumenta la utilización de la unidad central de procesamiento organizando los trabajos de manera que ésta siempre tenga algo que ejecutar. El sistema operativo escoge uno de los trabajos del depósito y comienza a ejecutarlo. En algún momento el trabajo tendrá que esperar, ya que el sistema ha pasado el control a otro programa y así sucesivamente. Mientras haya otro trabajo por ejecutar, la unidad central de procesamiento nunca estará inactiva. Dentro de los sistemas multiprogramados tenemos tres tipos:
 - **Tiempo compartido (Time Sharing).**- utiliza la planificación de la unidad central de procesamiento y la multiprogramación para proporcionar a cada usuario, que tiene su propio programa en memoria, una pequeña porción de un computador de tiempo compartido. La entrada/salida interactiva es demasiado lenta para un computador por lo que, para que la unidad central de procesamiento no permanezca inactiva, el sistema operativo la cambiará al programa de otro usuario. Esto ocurre tan rápidamente que cada usuario tiene la impresión de

que cuenta con su propia computadora, cuando en realidad todos lo comparten. Esta técnica fue propuesta por **Christopher Strachey**, para compartir el tiempo de una computadora entre varios trabajos, conmutando entre ellos de forma tan rápida que cada trabajo aparenta tener la computadora a su entera disposición.

- **Tiempo real.**- suele usarse como dispositivo de control en una aplicación dedicada. Tiene restricciones temporales bien definidas, por lo que el procesamiento debe llevarse a cabo dentro de los límites definidos o el sistema fallará. Se caracterizan por dar respuestas inmediatas.
 - **Combinados.**- es una mezcla de los dos anteriores. Aunque se ha intentado combinar la funcionalidad del tiempo compartido y el tiempo real en un solo sistema operativo, los resultados han sido pésimos debido a los obvios conflictos entre los requisitos de ambos tipos.
- **Sistemas distribuidos.**- un **sistema distribuido** es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora. es un sistema débilmente acoplado, es decir, los procesadores no comparten ni memoria, ni reloj, cada uno cuenta con su propia memoria local y se comunican a través de distintas líneas de comunicación. Los procesadores pueden variar de tamaño y función. Las principales ventajas son:
 - Compartición de recursos (Economía).
 - Aceleración de los cálculos (Velocidad).
 - Fiabilidad y confiabilidad.
 - Comunicación.
 - Crecimiento por incrementos.
 - **Sistemas Operativos paralelos.**- en estos tipos de Sistemas Operativos se pretende que cuando existan dos o más procesos que compitan por algún recurso se puedan realizar o ejecutar al mismo tiempo. En **UNIX** existe también la posibilidad de ejecutar programas sin tener que atenderlos en forma interactiva, simulando paralelismo (es decir, atender de manera concurrente varios procesos de un mismo usuario). Así, en lugar de esperar a que el proceso termine de ejecutarse (como lo haría normalmente), regresa a atender al usuario inmediatamente después de haber creado el proceso. Ejemplos de estos tipos de Sistemas Operativos están: **Alpha**, **PVM**, la serie **AIX**, que es utilizado en los sistemas **RS/6000** de **IBM**.
 - **Sistemas concurrentes.**- es un cuasi sinónimo de un sistema paralelo. Se utiliza para describir el acto de escribir un programa que ha de ejecutarse en paralelo.
 - **Sistemas Multitarea (Multitasking).**- es la ejecución concurrente de varias tareas, trabajos o procesos.
 - **Sistema monitor.**- es un sistema operativo que actúa como supervisor de un proceso o una serie de procesos, puede ser incluso un sistema operativo completo. Otra definición es la pensada por **Hoare**, para permitir el empleo compartido y controlado de recursos por procesos, de otro modo asíncrono, y que comprende la disposición del paso controlado de variables entre procesos.

1.4 Partes de un sistema operativo.

La interfase entre el sistema operativo y los programas de usuario está definida por el conjunto de **operaciones extendidas** que el sistema operativo ofrece. A ese conjunto de instrucciones se le conoce como **llamadas al sistema**.

Básicamente las llamadas al sistema son de dos tipos, las que se ocupan de los **procesos** y las que se ocupan del **sistema de archivos**.

Procesos.

Un **proceso** es un programa en ejecución. Cada proceso tiene un **espacio de direcciones**, es decir, una lista de posiciones en memoria. Contiene el programa ejecutable, los datos del programa y una pila. A cada proceso también se le asocia un conjunto de registros, que incluyen el **contador de programa (PC)**, el **apuntador de pila (SP)** y otros registros de hardware y toda la información para ejecutar el programa.