

## Unidad 1

### Sistemas Operativos. Tipos. Funciones

v 4.0.0 - 20100922

#### **INTRODUCCIÓN HISTÓRICA**

La evolución sufrida por los sistemas operativos ha venido provocada en la mayoría de los casos por cambios producidos en los componentes usados en la fabricación de los ordenadores y en el cambio en el diseño de su arquitectura; por este motivo veremos esta evolución en cada una de las etapas o "generaciones de ordenadores".

##### Primera generación (1945-1955): Válvulas de vacío

Durante esta primera etapa *los SO no existían*. El trabajo de activar o desactivar las unidades de E/S, revisión y corrección de errores, la introducción y el control de los programas los hacían manualmente los constructores de la máquina que eran los encargados del mantenimiento, así como de preparar las conexiones de circuitos adecuadas en cada caso, para cada programa, a través de un panel frontal que representaba mediante luces cualquier error producido durante la ejecución de éste. Llegado este caso, esta persona debía revisar la memoria y los registros para descubrir dónde se había producido el error.

El lenguaje de programación utilizado durante esta primera etapa era el *lenguaje máquina*, y los únicos dispositivos de E/S, las *tarjetas perforadas* y las impresoras.

Los programadores escribían sus programas en código máquina y después de ensamblar el programa objeto, lo montaban manualmente con otras funciones y unían después las tarjetas con los datos e introducían el paquete completo en la lectora. Eso sí, cuando les tocaba el turno, ya que tenían que solicitarlo apuntándose en unas listas de espera. Si durante la ejecución del programa no se producía ningún error, la salida era proporcionada a través de la impresora, pero en caso contrario debía volver al principio y, por supuesto, volver a esperar su turno. Este sistema de turnos a la hora de ejecutar los programas provocaba que, si una persona no consumía su tiempo, el ordenador estaba inactivo, o si un programador necesitaba más tiempo del asignado, debía volver a esperar.

Por otro lado, durante la ejecución de un programa sólo trabajaba la CPU, con lo que los dispositivos de E/S no podían ser utilizados.

Con el paso del tiempo se empezaron a realizar rutinas y funciones comunes que podían ser utilizadas por cualquier programador, montadores, cargadores y rutinas de E/S. Ese fue el primer paso hacia la creación de los SO. A esta forma de trabajo se la denominó "proceso serie".

##### Segunda generación (1955-1965): Transistores

Durante esta segunda etapa se produce una *diversificación laboral*, creándose distintos cometidos en el personal informático; los diseñadores de las máquinas no son las mismas personas que las fabrican. El personal de mantenimiento sólo se encarga del buen funcionamiento del ordenador. Aparecen los operadores, encargados de controlar el ordenador y las unidades de E/S. De esta forma, los programadores, una vez perforado su programa, lo unen a las tarjetas de datos y

pasan el paquete completo a los operadores, que se encargan de introducirlo en el ordenador.

Durante este periodo se perfeccionan los dispositivos de E/S de la etapa anterior y se unen a éstos las *cintas magnéticas*. También se incorpora el uso de ordenadores satélite para la gestión de las entradas y salidas.

Aparece en esta etapa el *lenguaje ensamblador* y el JCL (lenguaje de control de trabajo) en un primer periodo y, más tarde, *algunos de alto nivel como el FORTRAN*.

El proceso que se seguía era el siguiente: como ya se ha indicado, el programador perforaba las tarjetas correspondientes al programa y los datos e incluía entre ellas tarjetas de control que servían para diferenciar el programa de los datos, diferenciar un programa de otro, indicar el traductor necesario, etc. Una vez preparado, el paquete completo se le pasaba al operador, que se encargaba de introducir las tarjetas en el ordenador satélite. La función básica de este ordenador satélite era trasladar la información de dispositivos lentos (tarjetas) a dispositivos más rápidos (cintas magnéticas). De esta forma se preparaba una cinta que contenía una cola de trabajos con varios programas. Estos programas, además, tenían la posibilidad de utilizar bibliotecas de funciones comunes contenidas en cinta junto a los programas traductores que se montaban durante su paso por el satélite.

Esta cinta con la cola de trabajos se montaba después en el ordenador principal, más potente, que sólo leía y escribía sobre cinta (dispositivos rápidos). Una vez montada la cinta en este ordenador era necesaria la intervención de un programa (el primer SO), que era el encargado de ejecutar los programas y mandar las salidas a cinta. Esta cinta era montada posteriormente en el ordenador satélite, que pasaba los resultados a impresora o tarjeta.

Con esta forma de trabajo, denominada "sistema de procesamiento por lotes", "sistema batch", "lotes-serie" "cola serie" o también "tren de trabajos", la única manera de interactuar por parte del usuario era a través de las tarjetas de control y, evidentemente, el sistema de administración de la CPU utilizado era por monoprogramación, ya que hasta que no terminara un trabajo no se mandaba otro a ejecución.

Este primer SO, denominado "monitor", "supervisor" o también "ejecutivo", tenía las funciones de interpretar las tarjetas de control introducidas por el usuario y mandar a ejecución la rutina correspondiente a la tarjeta, y de cerrar el archivo de impresión correspondiente a un trabajo finalizado antes de dar paso al siguiente.

Estas funciones aumentaron y se perfeccionaron durante todo el periodo, incluyéndose más rutinas de control.

La estructura de este SO constaba de un módulo residente monitor, y de una parte no residente con rutinas que debían ser cargadas en memoria al principio de la ejecución del programa que las requiriese.

El monitor, al ser residente, debía estar protegido, impidiendo así que cualquier usuario pudiera acceder a él y que los programas no alterasen las posiciones de memoria que ocupaba.

Con el fin de protegerlo se toman dos medidas:

- La primera medida tiene como objetivo impedir que cualquier usuario acceda a él. Se crean dos estados del SO: el *estado superusuario*, donde se ejecutan las instrucciones llamadas privilegiadas sin antes hacer ninguna comprobación, y el *estado usuario*, donde está restringido el uso de estas instrucciones privilegiadas. Sin embargo, en algunos casos era necesaria la ejecución de este tipo de instrucciones en estado usuario. El caso se podía presentar cuando al estar ejecutándose una aplicación (modo usuario) se presentaba una instrucción de E/S. Para gestionarla era necesario pasar a modo superusuario y, así, el IOCS (sistema de control de entrada y salida) tomaría el control de ésta. Estamos hablando de, partiendo del modo usuario, pasar a modo superusuario, y esto sólo se podía hacer mediante una instrucción privilegiada que no se puede ejecutar en este modo (modo usuario). Para este tipo de situaciones se utilizaba el gestor de interrupciones, una parte del SO residente que controlaba los dos tipos de interrupciones existentes: las interrupciones hardware, asíncronas (no se sabía cuando podían ocurrir) y las interrupciones software, generadas desde los programas, únicas instrucciones privilegiadas que podían ejecutarse en modo usuario.

- La segunda medida que se toma es la creación de una unidad; la MMU (MemoryManagement Unit), que cuida que ningún programa acceda a una posición de memoria ocupada por la parte residente del SO, comprobando cualquier direccionamiento y rechazando los que invadan la zona correspondiente al módulo residente.

Con la aparición de los buffers de datos se va relegando la utilización de ordenadores satélites, ya que utilizando estas memorias intermedias se pueden solapar los procesos de E/S con la ejecución del programa.

### Tercera generación (1965-1980): circuitos integrados

Con la aparición de los circuitos integrados se empiezan a fabricar ordenadores más pequeños, de propósito general, que se comercializan y que disponen de un software compatible.

Al igual que en la etapa anterior se perfeccionan los dispositivos de E/S existentes y se empieza a utilizar el *disco magnético*. En un primer momento sólo lo utiliza el SO que lo utiliza como buffer. Recordemos que en la etapa anterior ya se dejaron de utilizar los ordenadores satélite y que se comenzó con la técnica del buffering, pues en esta etapa se comienza a utilizar otra técnica parecida aunque distinta, el spooling. Si con la primera se conseguía solapar la entrada y salida de un proceso con su ejecución, con la segunda se consigue solaparla entrada y salida de varios procesos con otro proceso distinto.

Los trabajos ya no se encuentran en una cinta, ahora se almacenan en el SPOOL (Simultaneous Peripheral Operation On Line) y el SO es el que planifica su secuenciación. Para ello divide la memoria en partes y mantiene así los distintos

procesos en ejecución, con esto la técnica de "lotes-serie" de la etapa anterior se consigue perfeccionar y evoluciona hacia una técnica de "lotes paralelos" o "secuencias múltiples". Pero ahora el problema se plantea porque la diferencia de velocidades entre el ordenador y los periféricos es mayor, con lo cual, si un proceso necesita realizar una operación de E/S, la CPU envía a ejecución otro trabajo. Si el SO consigue mantener en memoria una cantidad de trabajos suficiente, la CPU no estará inactiva.

Así se introduce la técnica de multiprogramación en los SO, que consigue aprovechar los tiempos en los que la CPU o los periféricos están inactivos, además de las zonas de memoria principal no ocupadas. La introducción de esta técnica también permite que el usuario vuelva a interactuar con el sistema, ya que con la técnica de multiprogramación en tiempo compartido el SO divide el tiempo de la CPU en pequeños intervalos denominados "rodajas" (time slices) que reparte para poder atender de forma concurrente a los distintos usuarios y a los distintos procesos que estuvieran en ejecución.

Como ya se ha indicado antes, para poder utilizar estas técnicas y así conseguir que el procesador esté trabajando continuamente, se necesita que la memoria sea capaz de almacenar varios trabajos. Para ello se introduce en los SO la técnica de memoria virtual, de forma que se consigue una memoria aparentemente mayor de lo que en realidad es, usando memoria de almacenamiento masivo en las que se apoya el SO para presentar al usuario una memoria central aparentemente mayor, ocupándose de las transferencias de datos y procesos de una memoria a otra sin que el usuario tenga que intervenir ni preocuparse por el tamaño limitado de la memoria central.

De la misma forma, el usuario tampoco tiene que preocuparse de qué periférico va a recibir la información que él envía, ya que durante esta etapa se introduce el concepto de "independencia de dispositivo". El usuario manda a imprimir y el SO le busca una impresora libre.

Se introduce también el concepto de "tiempo real", de forma que el ordenador da respuesta de forma inmediata a necesidades externas del sistema y se empiezan a fabricar los primeros multiprocesadores, con lo que el ordenador pasa a tener muchas más prestaciones. Todo esto lleva a los SO a tener que dar respuesta a estas máquinas con técnicas de multiproceso con las que deben mantener ocupados a varios procesadores.

#### Cuarta generación (1980-1995): ordenadores personales. Redes

Durante esta etapa comienza la conectividad de los equipos y se diseñan las primeras redes de ordenadores. Por este motivo comienzan a distinguirse dos nuevos tipos de SO, los sistemas operativos de red y los sistemas operativos distribuidos.

La conectividad aporta grandes ventajas en cuanto a la libertad de comunicación, pero también aporta inconvenientes imponiendo a los SO obligaciones y problemas relacionados con la seguridad, por ello se empieza a desarrollar la criptografía (cifrado de mensajes).

Los SO más importantes comercializados en esta etapa fueron el MSDOS para ordenadores personales, sencillo de utilizar y compatible; y el UNIX para grandes ordenadores, con características de multitarea y multiusuario.

Durante esta etapa y, en parte gracias a la compatibilidad de los SO, se produce un gran desarrollo de programas para ordenadores personales. Cabe destacar, entre el software desarrollado, la importancia que adquieren los sistemas de gestión de bases de datos en los que se apoyan directamente algunos SO.

#### Quinta generación (1995-): ordenadores superescalares. Redes internacionales

Los SO actuales tienen una concepción muy distinta a los de la primera etapa y esto es debido, sobre todo, a que la CPU del ordenador ya no es una máquina tan cara como en los primeros ordenadores, con lo cual ya no es tan importante que parte de su tiempo esté inactiva y, en segundo lugar, las CPU's actuales son componentes tan potentes que permiten interfaces gráficas, con lo que los SO, en la actualidad, prestan gran parte de su código para la comunicación con el usuario de una forma gráfica, visual y cada vez más amigable. Sin embargo, esto no ha impedido que SO monousuario incluyan técnicas de multiprogramación e incluso de multiproceso.

### **FUNCIONES ESPECÍFICAS DE LOS SISTEMAS OPERATIVOS**

*Un Sistema Operativo es un conjunto de programas que controla los recursos del ordenador y sirve para la ejecución de los programas e general convirtiéndose en el intermediario entre el hardware y el usuario. Tiene dos objetivos:*

Los principios básicos que deberán cumplir los SO:

- Abstracción: deberá estar diseñado de forma que las capas superiores de software no necesiten saber las características de las capas inferiores.
- Robustez: deberán ser sistemas donde el fallo de un programa no afecte al funcionamiento de otros, donde el usuario no pueda realizar ninguna acción que afecte al buen funcionamiento del sistema y donde en el caso de producirse algún error, sin que el sistema llegue a colapsarse, éste sea claramente indicado y documentado, así como expresadas de forma clara las posibles decisiones que se deben tomar.
- Estandarización: actualmente existen numerosas plataformas distintas entre las que un usuario debe elegir, y es necesario elegir bien, ya que es costoso y lento cambiar de una a otra. Por ese motivo se debe tender a la estandarización, de manera que el sistema sea capaz de adaptarse a los avances tecnológicos, incluso estar diseñado de forma que parte de éste pueda ser actualizado constantemente para reflejar los posibles cambios.
- Facilidad de uso y de desarrollo de software: serán fáciles de implementar a la vez que deberán facilitar el trabajo a los programadores.

Las funciones específicas de un SO:

1. Comunicación con el usuario.
2. Control y gestión de los recursos físicos.
3. Gestión de usuarios.
4. Ejecución de programas (Gestión de procesos).
5. Gestión de la información. (Sistema de Ficheros)
6. Comunicación entre sistemas.

## **TIPOS DE SISTEMAS OPERATIVOS**

Hay una gran variedad, en función de sus características, y cada uno se caracteriza por las técnicas de acceso a la información y el modo de su proceso.

Según el *número de usuarios*:

Monousuario: Son aquellos SO que soportan solamente un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que dicho usuario pueda ejecutar en un mismo instante de tiempo. Solo hay un usuario accediendo a los recursos del sistema y hasta que este usuario no termine, no se liberan los recursos.

Multiusuario: Son aquellos SO capaces de dar servicio a más de un usuario simultáneamente, bien sea por medio de varias terminales conectadas a la computadora, bien sea mediante sesiones remotas en un red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario pueda ejecutar simultáneamente.

Según el *número de procesos*

Monotarea: Son aquellos SO que permiten una sola tarea a la vez.

Multitarea: Son aquellos SO que permiten al usuario realizar varias tareas simultáneamente. Existen varios tipos de multitareas. Un tipo sencillo de multitarea es la *conmutación de contextos*, en la que dos o más aplicaciones se cargan al mismo tiempo, pero solo se está procesando la aplicación que se encuentra en primer plano; es decir, la que ve el usuario. Para activar otra tarea que se encuentre en segundo plano, el usuario tiene que traer al primer plano la ventana que tenga esa aplicación. En *multitarea cooperativa* (utilizada por el SO Macintosh) las tareas en segundo plano reciben tiempo de procesamiento durante los tiempos muertos de la tarea que se encuentra en primer plano. En los sistemas multitarea de *tiempo compartido*, como el SO OS/2 cada tarea recibe la atención del microprocesador durante una fracción de segundo. Para mantener el sistema en orden, cada tarea recibe un nivel de prioridad.

Según el *número de procesadores*

Monoprocesador: Es aquel SO capaz de manejar solamente un procesador de la computadora. En caso de que dicha computadora dispusiese de más de un procesador, el segundo estaría inutilizado. El ejemplo más típico de este tipo de SO es el DOS.

Multiprocesador: Un SO multiproceso es aquel capaz de usar más de un procesador del sistema de manera simultánea, con objeto de distribuir su carga de trabajo. Estos SO pueden trabajar de dos formas: simétrica o asimétrica. Cuando se trabaja de forma asimétrica, el sistema operativo selecciona a uno de los procesadores del sistema que actúa como procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos. Cuando se trabaja de forma simétrica, los procesos o partes de ellos (threads) son enviados indistintamente a cualquiera de los procesadores disponibles; por tanto, en esta forma de trabajo se tiene al menos teóricamente una mejor distribución y equilibrio en la carga de trabajo. Un thread es la parte activa en memoria y avanzando de un proceso. Un aspecto importante que hay que considerar en estos sistemas es la forma de crear aplicaciones para un mejor aprovechamiento de los diversos procesadores.

#### *Orientados a la red:*

Sistemas operativos de red: Son aquellos SO que tienen la capacidad de interactuar con SO de otras computadoras a través de un medio de transmisión, con el fin de poder intercambiar información, permitir la transferencia de ficheros o poder ejecutar comandos remotos, como actividades más importantes. Para poder ejecutar estas operaciones, el usuario ha de conocer un conjunto de comandos o llamadas al sistema, así como la ubicación de los diferentes recursos a los que el quiera acceder.

Sistemas operativos distribuidos: Los SO distribuidos abarcan los servicios de los SO de red, logrando la integración de recursos como memoria, procesos, unidades centrales de proceso, impresoras, etcétera, en una sola máquina virtual cuyo acceso es totalmente transparente para el usuario. En esta situación el usuario no necesita conocer la ubicación de los diferentes recursos, sino que los conoce por su nombre y los utiliza como si dichos recursos estuviesen en su lugar de trabajo habitual. Es un solo sistema operativo virtual distribuido por varias máquinas interconectadas entre sí.

## **SUBSISTEMAS DE UN SISTEMA OPERATIVO**

En los sistemas se establecen cuatro niveles de protección en los que hay que establecer unas medidas de seguridad para evitar el enfrentamiento y acceso indebido a determinadas posiciones de memoria e información. Los niveles son:

Protección de memoria Consiste en que un trabajo no puede acceder a otro. Una de las técnicas para evitar esto es la utilización de registros frontera y detección por hardware.

Protección de la CPU: Para que todas las tareas cuenten con el mismo tiempo se utilizan temporizadores que aseguran una cantidad de tiempo para cada usuario activo.

Protección de E/S: Se establece una protección a la hora de compartir recursos para operaciones de E/S. La compartición de recursos en sistemas de multiprogramación la establece el SO.

Protección de los datos: La pérdida de datos puede venir motivada por alguno de los siguientes aspectos:

*Errores de software y errores de hardware:* Como por ejemplo errores de programas, fallos de disco, fallos del microprocesador, fallos en la comunicaciones, etc..

*Errores motivados por el usuario:* borrado voluntario o involuntario de software, etc...

*Pérdidas debidas a catástrofes:* inundaciones, incendios, etc...

La mayoría de las pérdidas de datos pueden solucionarse si se realizan copias de seguridad y que deberían guardarse en lugares diferentes a los destinados a los datos originales.