

## Unidad I: Introducción a los sistemas operativos

### 1.1 Definición y concepto

El sistema operativo es el programa (o software) más importante de un ordenador. Para que funcionen los otros programas, cada ordenador de uso general debe tener un sistema operativo. Los sistemas operativos realizan tareas básicas, tales como reconocimiento de la conexión del teclado, enviar la información a la pantalla, no perder de vista archivos y directorios en el disco, y controlar los dispositivos periféricos tales como impresoras, escáner, etc. En sistemas grandes, el sistema operativo tiene incluso mayor responsabilidad y poder, es como un policía de tráfico, se asegura de que los programas y usuarios que están funcionando al mismo tiempo no interfieran entre ellos. El sistema operativo también es responsable de la seguridad, asegurándose de que los usuarios no autorizados no tengan acceso al sistema. Clasificación de los Sistemas Operativos

Los sistemas operativos pueden ser clasificados de la siguiente forma:

- Multiusuario: Permite que dos o más usuarios utilicen sus programas al mismo tiempo. Algunos sistemas operativos permiten a centenares o millares de usuarios al mismo tiempo.
- Multiprocesador: soporta el abrir un mismo programa en más de una CPU.
- Multitarea: Permite que varios programas se ejecuten al mismo tiempo.
- Multigrano: Permite que diversas partes de un solo programa funcionen al mismo tiempo.
- Tiempo Real: Responde a las entradas inmediatamente. Los sistemas operativos como DOS y UNIX, no funcionan en tiempo real.

Cómo funciona un Sistema Operativo Los sistemas operativos proporcionan una plataforma de software encima de la cual otros programas, llamados aplicaciones, puedan funcionar. Las aplicaciones se programan para que funcionen encima de un sistema operativo particular, por tanto, la elección del sistema operativo determina en gran medida las aplicaciones que puedes utilizar. Los sistemas operativos más utilizados en los PC son DOS, OS/2, y Windows, pero hay otros que también se utilizan, como por ejemplo Linux. Cómo se utiliza un Sistema Operativo Un usuario normalmente interactúa con el sistema operativo a través de un sistema de comandos, por ejemplo, el sistema operativo DOS contiene comandos como

copiar y pegar para copiar y pegar archivos respectivamente. Los comandos son aceptados y ejecutados por una parte del sistema operativo llamada procesador de comandos o intérprete de la línea de comandos. Las interfaces gráficas permiten que utilices los comandos señalando y pinchando en objetos que aparecen en la pantalla.

## **1.2 Funciones y características**

-Gestiona los recursos del ordenador en sus niveles más bajos. -Dispone de una interface (elemento que hace posible la fácil comunicación usuario maquina) liberando al usuario del conocimiento del hardware. El SO Windows se basa en una interface gráficas, "GUI" (Interface Gráfica de Usuario), permitiendo al usuario interactuar con el hardware de una forma sencilla y rápida. -Sobre el SO funcionan el resto de programas y aplicaciones del software.

Sus generaciones:

-Generación 0º: década de los 40 los sistemas informáticos no disponían de SO con lo que los usuarios de estos debían introducir las instrucciones en código binario lo que hacía su uso restringido a personas de mucho conocimiento en esa materia. -Generación 1º: década de los 50. Aparece el primer SO para lograr la fluidez en la trasmisión de información. Aparece el JLC (lenguaje de control de trabajo), se usaban tarjetas perforadas y eran controladas por operadores (personas con cierto conocimiento). Posteriormente se pasaron de las tarjetas a las cintas perforadas y estas iban mucho más rápido. En el primer SO había en ocupaba en memoria 64 KB (bastante en función de la capacidad total de la memoria en aquella época).

Grafico interno de un sistema operativo de la época y en parte en la actualidad

Generación 2º: Se dan los primeros pasos de la multiprogramación es decir varios programas de usuario. El SO reparte tiempos del procesador. Aparece la llamada tecnología DMA usa buffers entre terminales: impresora, etc. ¿Qué es un buffer?:

Es una fuente de almacenamiento temporal que reside en el propio dispositivo ya sea de entrada, o de salida. Aparece el término spooling: viene a ser como lo que gestiona la cola de instrucciones en el buffer.

### **1.3 Evolución histórica**

La primera computadora digital real fue diseñada por el matemático inglés Charles Babbage (1792 - 1871). Aunque Babbage gastó la mayor parte de su vida y de su fortuna intentando construir su “ maquina analítica “ , nunca la hizo funcionar adecuadamente porque era un diseño puramente mecánico y la tecnología de su época no podía producir las ruedas, el engranaje, levas y otras partes mecánicas con la alta precisión que el necesitaba. Sin tener que decirlo, la maquina analítica no tuvo un sistema operativo.

La primera generación (1945 - 1955): Tubos de vacío y tableros enchufarles

Después de los esfuerzos frustrados de Babbage, se progresó poco en la construcción de computadoras digitales hasta la segunda guerra mundial, alrededor de la mitad de la década de 1940, Howard Arikuen en Harvard, Jon Von Neumann en el Instituto de Estudios Avanzados en Princeton, J. Pres per Esker y William Macuche en la Universidad de Pennsylvania y Konrad Cuse en Alemania, entre otros, todos obtuvieron resultados óptimos en la construcción de máquinas de cálculo mediante el uso de tubos de vacío.

En estos primeros días, un grupo singular de personas diseño, construyo, programo, opero y dio mantenimiento a cada máquina. Toda la programación se realizó en lenguaje de maquina absoluto. Los lenguajes de programación se desconocían (todavía no existía el lenguaje ensamblador). Los primeros sistemas operativos eran extraños. El modo usual de operación consistía en que el programador firmaba para tener acceso a un bloque de tiempo en la hoja de registro situada en la pared, después bajaba al cuarto de máquinas, insertaba su

tablero enchufarle en la computadora y pasaba las siguientes horas esperando que ninguno de los 20,000 tubos de vacío se fundiera durante la ejecución de su programa.

Al inicio de la década de 1950, la rutina había mejorado un poco con la introducción de la tarjetas perforadas. Ahora era posible escribir en tarjetas y leerlos, en vez de utilizar tableros enchufarles; de lo contrario el procedimiento era el mismo.

La segunda generación (1955 - 1965): Transistores y sistemas de lote

La introducción del transistor a mediados de la década de 1950 cambió la imagen radicalmente. Las computadoras se volvieron lo suficientemente confiables, en un principio hubo una clara separación entre los diseñadores, armadores, operadores, programadores y personal de mantenimiento. Estas máquinas se instalaban en cuartos de computadoras especialmente acondicionados con aire, con cuerpo de operadores profesionales para accionarlas. Un programador primeramente escribiría el programa en papel (en FORTRAN o en lenguaje Ensamblador) y después lo perforaría en tarjetas. Después llevaría la pila de tarjetas al cuarto de introducción al sistema y la entregaría a uno de los operadores el cual iniciaba el proceso en la computadora, este proceso desperdiciaba mucho tiempo.

Dado el alto costo del equipo, no es sorprendente que las personas buscaran rápidamente maneras de reducir el tiempo perdido. La solución que generalmente se adoptaba era el sistema de lote. La idea implícita en este sistema era la de conjuntar un cajón lleno de trabajos en el cuarto de introducción al sistema y después leerlos en una cinta magnética mediante el uso de una computadora (relativamente) pequeña y poco costosa, como la IBM 1401.

Después de casi una hora de recolectar un lote de trabajos, la cinta se volvía a enrollar y se llevaba al cuarto de máquinas. Después el operador cargaba un

programa especial ( el ancestro del sistema operativo de hoy en día ) , el cual leía el primer trabajo y lo ejecutaba, la salida se escribía en una segunda cinta, en vez de imprimirse. Después de terminar cada trabajo, el sistema operativo leía automáticamente el siguiente trabajo de la cinta, y comenzaba a ejecutarlo.

La estructura de un trabajo de entrada común arrancaba con una tarjeta \$JOB, que especifica el tiempo máximo de ejecución en minutos, el número de cuenta que se cargara y el nombre del programador. Después venía una tarjeta \$FORTRAN, que indicaba al sistema operativo que debía cargar el compilador de FORTRAN. Venía seguido de un programa que debía compilarse y después de una tarjeta \$LOAD, que ordenaba al sistema operativo cargar el programa objeto recién compilado, después venía la tarjeta \$RUN, que indicaba al sistema operativo que debía ejecutar el programa con los datos que le seguían. Por último, la tarjeta \$END marcaba el final del trabajo. Los sistemas operativos comunes eran FMS (el sistema monitor del FORTRAN) e IBSYS, sistema operativo de IBM de la 7094.

La tercera generación (1965 - 1980): Circuitos integrados (CI) y multiprogramación  
Al inicio de la década de 1960 muchos fabricantes de computadoras tenían dos líneas de trabajo distintas y totalmente incompatibles. Por un lado existían las computadoras científicas de grande escala orientadas a las palabras, como la 7094, que se utilizaban para realizar cálculos numéricos de ciencias e ingeniería. Por el otro lado estaban las computadoras comerciales orientadas a los caracteres, como 1401, que se utilizaban para el ordenamiento de cintas e impresión por parte de bancos y compañías de seguros.

El desarrollo y mantenimiento de dos líneas de productos diferentes era una proposición costosa para los fabricantes. Además, muchos nuevos compradores de computadoras necesitaban una maquina pequeña, pero después se expandían y querían una máquina de mayor tamaño que ejecutara todos sus programas antiguos, pero con mayor velocidad.

IBM intento resolver estos dos problemas de un solo golpe introduciendo en el mercado el Sistema/360. El 360 era una serie de máquinas compatibles con el software que variaban del tamaño de la 1401 a una mucho más poderosa que la 7094, el 360 estaba diseñado para realizar cálculos tanto científicos como comerciales. Por lo tanto una sola familia de máquinas podía satisfacer las necesidades de todos los clientes.

El sistema 360 fue la primera línea importante de computadoras que utilizo circuitos integrados (CI), con lo cual ofreció una mayor ventaja de precio/rendimiento sobre las máquinas de la segunda generación.

La intención era que todo el software, como el sistema operativo, tenía que funcionar en todos los modelos. Tenía que correr en sistemas pequeños, y en sistemas muy grandes. Tenía que funcionar adecuadamente en sistemas con algunos periféricos y en sistemas con muchos periféricos.

No había manera de que IBM escribiera una pieza de software que cumpliera todos esos requisitos conflictivos. El resultado fue un sistema operativo enorme y extraordinariamente complejo. Constaba de millones de líneas de lenguaje ensamblador escritas por miles de programadores, y contenía miles y miles de errores ocultos.

A pesar de tamaño y problemas enormes, OS/360 y los sistemas operativos similares de la tercera generación satisfacían a muchos de sus clientes razonablemente bien, También popularizaron varias técnicas importantes ausentes en los sistemas operativos de la segunda generación. La más importante de estas fue la multiprogramación. Cuando el trabajo corriente se detenía para esperar a que se completara una operación en cinta u otra operación de E/S, la unidad central de procesamiento (CPU) simplemente permanecía ociosa hasta que terminara la operación de E/S. La solución que evoluciono consistía en partir la memoria en varias partes, con trabajo diferente en cada partición. Mientras que un trabajo esperaba a que se completara la E/S, otro trabajo podía estar utilizando

la CPU. Si se podían mantener suficientes trabajos en la memoria central al mismo tiempo, la CPU podía mantenerse ocupada casi el 100% del tiempo.

Otra característica de importancia en los sistemas operativos de la tercera generación era la capacidad de leer trabajos de tarjetas contenidas en el disco tan pronto como se llevaban al cuarto de computación. Siempre que se terminaba un trabajo, el sistema operativo podía cargar uno nuevo del disco en la partición no vacía y ejecutarlo, esta técnica se denomina manejo por cola de impresión.

Los sistemas operativos de la tercera generación seguían siendo básicamente sistemas de lote. Con los sistemas operativos de la tercera generación, el tiempo entre la entrega de un trabajo y la devolución de la salida comprendía a menudo varias horas.

El deseo de obtener un tiempo de respuesta corto marco el camino para el tiempo compartido, variante de la multiprogramación, en la cual cada usuario tiene una terminal en línea. En un sistema de tiempo compartido si hay 20 usuarios dentro del sistema y 17 de ellos están pensando o platicando o bien tomando café, la CPU puede distribuirse en turno para los tres trabajos que necesitan servicio. Aunque el primer sistema de tiempo compartido (CTSS) serio fue creado en MIT en una unidad 7094 especialmente modificada, no se volvió popular sino hasta que el hardware de protección necesario se diseminó durante la tercera generación.

Después del éxito del sistema CTSS, MIT, Bell laboratorios y General electric decidieron embarcarse en el desarrollo de la “computadora de servicio público “. Conocido como MULTICS (Multiplexada Information and computan Services, información multicanal izada y servicio de computación). Para resumir una larga historia, MULTICS introdujo muchas ideas originales en la literatura de computación, pero su construcción era más difícil de lo que nadie había sospechado. MULTICS tuvo enorme influencia sobre otros sistemas subsiguientes. Otro avance durante la tercera generación fue el crecimiento de las

minicomputadoras, comenzando con DEC PDP-1 en 1961. Uno de los científicos que había trabajado en el proyecto MULTICS, Ken Thompson, halló después una pequeña PDP-7 y empezó a escribir después una versión desguarnecida de MULTICS para un usuario. Este sistema se llamó "UNICS" (Uniplexed Information and computan Services, información en canalizada y servicio de computación), pero su ortografía cambió más tarde por UNIX. UNIX se ha desplazado a más computadoras que ningún otro sistema operativo de la historia y su uso sigue aumentando rápidamente.

#### La cuarta generación (1980 - 1990): Computadoras personales

Con la creación de los circuitos integrados LSI (integración a grande escala), chips que contiene miles de transistores en un centímetro cuadrado de silicón, la era de computadora personal vio sus inicios. Dos sistemas operativos han dominado la escena de la computadora personal: MS-DOS, escrito por Microsoft, Inc., para la IBM PC y otras computadoras que utilizan la CPU Intel 8088 y sus sucesores. Y UNIX, que domina en las computadoras personales mayores que hacen uso de CPU Motorola 68000. Aunque la versión inicial de MS-DOS era relativamente primitiva, versiones subsiguientes han incluido más y más características de UNIX, lo que no es totalmente sorprendente dado que Microsoft es un proveedor importante de UNIX, que usa el nombre comercial de XENIX. Un avance importante que empezó a tomar su sitio a mediados de la década de 1980 es el desarrollo de redes de computadoras personales que corren sistemas operativos en red y sistemas operativos distribuidos. En un sistema operativo en red, los usuarios tienen conocimiento de la existencia de múltiples computadoras y pueden ingresar en máquinas remotas y reproducir archivos de una máquina a la otra. Cada máquina ejecuta su sistema operativo local y tiene un usuario propio (o usuarios). Un sistema distribuido, es aquel que se presenta ante sus usuarios como un sistema un procesador tradicional, aunque en realidad este compuesto de múltiples procesadores. En un sistema distribuido real, los usuarios no tienen conocimiento de donde se están ejecutando sus programas o de donde están ubicados sus archivos; todo esto se debe manejar en forma automática y eficiente por medio del sistema operativo. Los sistemas operativos en red no son



fundamentalmente diferentes de los sistemas operativos unos procesadores. Sin duda necesitan un controlador de interfaz en red y algún software de bajo nivel para impulsarlo, así como programas para lograr un ingreso remoto al sistema y un acceso remoto del archivo. Los sistemas operativos distribuidos reales requieren más que simplemente agregar un poco de código a un sistema operativo un procesador, ya que los sistemas operativos distribuidos y centralizados difieren de manera decisiva.

#### Historia de Minis

Cuando UNIX era joven (versión 6), el código fuente se encontraba en todas partes, con autorización de AT&T, y se estudiaba frecuentemente, John Linos, llegó a escribir un pequeño folleto que describía su operación, línea por línea, este folleto se utilizó como libro de texto en muchos cursos universitarios. Cuando AT&T entregó la versión 7, empezó a comprender que UNIX era un valioso producto comercial, así que emitió la versión 7 con una licencia que prohibía el estudio del código fuente en cursos con el objeto de evitar poner en peligro su condición como secreto comercial, muchas universidades se quejaron simplemente descartando el estudio de UNIX y enseñando solo teoría. Por desgracia, el solo enseñar teoría deja al estudiante con una visión desproporcionada de lo que en realidad es un sistema operativo. Para remediar esta situación, decidí escribir un nuevo sistema operativo que sería compatible con UNIX desde el punto de vista del usuario, pero completamente diferente en el interior. El nombre MINIX surge de mini-UNIX porque es lo suficientemente pequeño que hasta alguien que no sea maestro puede entender la forma en que trabaja. MINIX tiene otra ventaja sobre UNIX, se escribió una década después que UNIX y se ha estructurado en forma más modular. El sistema de archivo de MINIX, por ejemplo, no es parte del sistema operativo en absoluto, pero corre como un programa de usuario. Otra diferencia es que UNIX se diseñó para ser eficiente; MINIX se diseñó para ser legible, el código de MINIX, por ejemplo, tiene más de 3000 comentarios en él. MINIX se ha diseñado para ser compatible con la versión 7 de UNIX. AL igual que UNIX, MINIX se escribe en lenguaje de programación C.

La implementación inicial se hizo en la IBM PC, MINIX no requiere un disco duro para correr, con lo cual se ajusta a los presupuestos de muchos estudiantes

## 1.4 Clasificación

**Sistemas Operativos de multitarea:** con este sistema operativo es que la computadora procesa diferentes tareas al mismo tiempo. Existen varios tipos de multitareas. Estos sistemas Se distinguen por sus habilidades para poder soportar la ejecución de dos o más trabajos activos (que se están ejecutando) al mismo tiempo. Esto trae como resultado que la Unidad Central de Procesamiento (UCP) siempre tenga alguna tarea que ejecutar, aprovechando al máximo su utilización.

Los sistemas operativos que soportan estas multitareas son: UNIX, Windows 95, Windows 98, Windows NT, MAC-OS, OS/2.

**Sistema Operativo Monotareas:** los sistemas operativos monotareas son más primitivos y es todo lo contrario a los de multitareas es decir, solo pueden manejar un proceso en cada momento o que solo puede ejecutar las tareas de una en una. Por ejemplo cuando la computadora está imprimiendo un documento, no puede iniciar otro proceso ni responder a nuevas instrucciones hasta que se termine la impresión. El sistema operativo que soporta el monotareas es DOS.

**Sistema Operativo Monousuario:** Los sistemas monousuarios son aquellos que nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se esté ejecutando. Estos tipos de sistemas son muy simples, porque todos los dispositivos de entrada, salida y control dependen de la tarea que se está utilizando, esto quiere decir, que las instrucciones que se dan, son procesadas de inmediato; ya que existe un solo usuarios. Los sistemas operativos que soportan el monousuario son DOS e incluso en el actual Windows XP y sus predecesores.

**Sistema Operativo Multiusuario:** Es todo lo contrario a monousuario; y en esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este tipo de sistemas se emplean especialmente en redes.

### 1.5 Estructura: niveles o estratos de diseño

**Nivel 1. Gestión de Memoria:** Proporciona las facilidades de bajo nivel para la gestión de memoria secundaria necesaria para la ejecución de procesos.

**Nivel 2. Procesador:** Se encarga de activar los quantums de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.

**Nivel 3. Entrada/Salida:** Proporciona las facilidades para poder utilizar los dispositivos de E/S requeridos por los procesos.

**Nivel 4. Información o Aplicación o Intérprete de Lenguajes:** Facilita la comunicación con los lenguajes y el sistema operativo para aceptar las órdenes en cada una de las aplicaciones. Ejecutando un programa el software de este nivel crea el ambiente de trabajo e invoca a los procesos correspondientes.

**Nivel 5. Control de Archivos:** Proporciona la facilidad para el almacenamiento a largo plazo y manipulación de archivos con nombre, va asignando espacio y acceso de datos en memoria.

El núcleo o cerner realiza diferentes funciones tales como:

- Manejo de interrupciones.
- Creación y destrucción de procesos.
- Cambio de estado de los procesos.
- Despacho
- Suspensión y reanudación de procesos.
- Sincronización de procesos.
- Comunicación entre procesos.

- Manipulación de los bloques de control de procesos.
- Apoyo para las actividades de entrada/salida.
- Apoyo para asignación y liberación de memoria.
- Apoyo para el sistema de archivos.
- Apoyo para el mecanismo de llamada y retorno de un procedimiento.
- Apoyo para ciertas funciones de contabilidad del sistema.

## **1.6 Núcleo**

Una definición más específica de lo que es el núcleo ( Kernel ) de un sistema operativo es: un conjunto de rutinas cuya misión es la de gestionar el procesador, la memoria, la entrada/salida y el resto de procesos disponibles en la instalación. Toda esta gestión la realiza para atender al funcionamiento y peticiones de los trabajos que se ejecutan en el sistema.