

# SISTEMAS OPERATIVOS

## ¿Qué son?

- Software de comunicación entre el dispositivo y el usuario, ofreciéndole un ambiente amigable y sencillo de interpretar
- Controlador de hardware del sistema (soporte lógico que controla el funcionamiento del equipo físico).
- Desde el punto de vista del usuario, es un conjunto de programas y funciones que ocultan los detalles del hardware, ofreciendo al usuario una vía sencilla y flexible de acceso al mismo.
- Programa específico o software de base que nos brinda un entorno para que podamos conectarnos e interactuar con el hardware de la computadora

## Recursos administrados por el sistema operativo

- Gestionar la memoria de acceso aleatorio y ejecutar las aplicaciones, designando los recursos necesarios
- Administrar la CPU, gracias al algoritmo de programación
- Direccionar las entradas y salidas de datos (a través de drives), por medio de los periféricos de entrada y salida
- Administrar la información para el buen funcionamiento de la PC
- Dirigir las autorizaciones de uso para el usuario
- Administrar los archivos

## Ejemplo de uso

- Aparatos de uso doméstico
  - Cuando accedemos a una app desde nuestro celulares, estamos usando la interfaz del sistema operativo (ejem: android) porque es este quien le da las órdenes al procesador para que lo ejecute
  - El procesador es informado de cosas sencillas como mover el mouse u oprimir una tecla por medio del sistema operativo

## Clases

- De uso doméstico
- Servidores
  - Los SO son heredados del sistema UNIX, como Red Hat, Windows Server o Debian

## Sistemas operativos según licencia

- Open Source (libre)
  - Permiten modificar, usar y adaptar un sistema operativo
    - Ejemplo: Ubuntu y Red Hat
- Proprietary software (privativo)
  - Son de propietarios y no permiten modificaciones
    - Ejemplo: Windows

## Tipos de sistema operativo

Los SO varían según el hardware y la función de cada dispositivo

- **Gestión de usuario**
  - Multiusuario [servidores]
    - SO que permite que varios usuarios ejecuten simultáneamente sus programas [conectados al mismo núcleo]
      - Se usa en las computadoras mainframe
        - Una **unidad central** (en inglés **mainframe**), es una computadora utilizada principalmente por grandes organizaciones para aplicaciones críticas, procesamiento de datos masivos (como censos y estadísticas de la industria y del consumidor, planificación de recursos empresariales y transacciones a gran escala procesamiento). Una computadora central es más grande y tiene más potencia de procesamiento que algunas otras clases de computadoras, como miniordenadores, servidores, estaciones de trabajo y computadoras personales.
    - Ejem:
      - Windows (a partir de XP)
        - Está diseñado para permitir la configuración de múltiples usuarios e iniciar varias sesiones al mismo tiempo
      - Unix
      - Linux
      - Mac OSX
      - Solaris
      - Android (en tablets)
  - Monousuario [uso doméstico]
    - SO que solamente permite ejecutar los programas de un usuario a la vez
      - Como aceptar la descarga de una actualización
    - Ejem:

- Windows (hasta Me)
  - DOS
  - Android (en celulares)
- **Gestión de tareas**
  - Multitarea
    - SO que puede ejecutar varios procesos al mismo tiempo
    - Ejem:
      - Windows
        - Cada usuario abre sus tareas de manera independiente
      - Unix
      - Linux
      - Mac OSX
  - Monotarea
    - SO que solamente permite ejecutar un proceso a la vez sin que se pueda interrumpir
    - Ejem:
      - DOS
      - Windows (Me)
- **Gestión de recursos**
  - Centralizado
    - SO que solo permite utilizar los recursos de un solo ordenador
    - Ejem:
      - Windows
      - Linux
      - Mac OSX
      - Unix
  - Distribuido
    - SO que permite ejecutar los procesos de más de un ordenador al mismo tiempo
    - Distribuyen sus tareas en múltiples CPUs
    - Ejem:
      - Novell Netware
      - Windows Server
      - Cisco IOS
      - Unix
      - Linux
- **Cliente-servidor**
  - Este sistema es el más reciente de todos.
  - Sirve para toda clase de aplicaciones, por lo tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.
  - La idea es mantener la visión que tiene el usuario de un computador personal, pero la red le permite compartir el espacio del disco o la impresora con el fin de economizar los recursos.

- Estos SO presentan una desventaja y es que no resuelven los problemas de compartir información, lo que dificulta el desarrollo en grupo
- **Estructura interna SO**
  - Monolítica
    - Constituido por un solo programa, compuesto de una serie de rutinas entrelazadas entre sí, de modo que puedan comunicarse entre ella
    - Suelen estar hechos a medida, por lo que son muy rápidos, pero no tienen flexibilidad para soportar diferentes tipos de aplicaciones
    - Ejem:
      - VMS
      - Linux
      - Multics
      - Windows (hasta Me)
  - Jerárquica
    - Una parte del sistema contiene subpartes y está organizado en forma de niveles
    - Subdivide en capas o anillos perfectamente definidos y con una clara interfaz con respecto al resto de los recursos
    - Ejem:
      - Unix
      - Multics
  - Máquina virtual
    - Separan dos conceptos que suelen estar unidos en el resto de los sistemas: multiprogramación y máquina extendida
    - Integra distintos SO dando la sensación de ser varias máquinas diferentes
    - Ejem:
      - Microsoft Hyper-V
      - VMware
      - VirtualBox
      - QEMU
      - Kernel-Based Virtual machine

## **Clasificación y comparación**

- No todos los hardware están en la capacidad de soportar o requerir el mismo SO
  - Ejem:
    - Computadora con menos de 4Gb de RAM difícilmente soportará un SO de 64 bits

## Generaciones de sistemas operativos

- Generación cero
  - Década de 1940
  - Las computadoras electrónicas digitales no tenían sistema operativo.
  - Los programas, por lo regular, manejaban un bit a la vez, en columnas de switches mecánicos.
  - Los programas de lenguaje máquina manejaban tarjetas perforadas.
- Primera generación
  - 1945-1955
  - Tubos de vacío y tableros enchufables
    - Se lograron construir máquinas calculadoras usando tubos de vacío. Estas máquinas eran enormes y ocupaban cuartos enteros con decenas de miles de tubos de vacío, pero eran mucho más lentas que incluso las computadoras personales más baratas de la actualidad.
    - Toda la programación se realizaba en lenguaje de máquina absoluto.
- Segunda generación
  - 1955-1965
  - Transistores y sistemas de lote
    - Estas máquinas se encerraban en cuartos de computadora con acondicionamiento de aire especial. Para ejecutar un programa, un programador escribía primero el programa en papel (en FORTRAN o ensamblador) y luego lo perforaba en tarjetas. Después, llevaba el grupo de tarjetas al cuarto de entrada y lo entregaba a uno de los operadores. Cuando la computadora terminaba el trabajo que estaba ejecutando en ese momento, se separaba la salida impresa y se llevaba al cuarto de salida donde el programador podía buscarla. Luego, el operador tomaba uno de los grupos de tarjeta traídos del cuarto de entrada y lo introducía en el lector. Si se requería el compilador de FORTRAN, el operador tenía que traerlo de un archivero e introducirlo en el lector.
    - Dado el alto costo del equipo, la solución que se adoptó generalmente fue el sistema por lotes. El principio de este modo de operación consistía en juntar una serie de trabajos en el cuarto de entrada, leerlos y grabarlos en una cinta magnética usando una computadora pequeña y (relativamente) económica.
    - Después de cerca de una hora de reunir un lote de trabajos, la cinta se rebobinaba y se llevaba al cuarto de la máquina, donde se montaba en una unidad de cinta. El operador cargaba entonces un programa especial, que leía el primer trabajo de la cinta y lo ejecutaba. La salida se escribía en una

segunda cinta, en lugar de imprimirse. Cada vez que terminaba un trabajo, el sistema operativo leía automáticamente el siguiente trabajo de la cinta y comenzaba a ejecutarlo.

- Tercera generación
  - 1965-1970
  - Circuitos integrados ( CI ) y multiprogramación
    - Las máquinas diferían solo en el precio y el rendimiento (memoria máxima, velocidad del procesador, número de dispositivos de E/S permitidos, entre otros). IBM trató de resolver simultáneamente ambos problemas introduciendo la System/360, puesto que todas las máquinas tenían la misma arquitectura y conjunto de instrucciones, los programas escritos para una máquina podían ejecutarse en todas las demás, al menos en teoría.
    - Los 360 y los sistemas operativos de tercera generación parecidos a él, producidos por otros fabricantes de computadoras, lograron satisfacer a sus clientes en un grado razonable y también popularizaron varias técnicas clave que no existían en los sistemas operativos de la segunda generación. Tal vez la más importante de ellas haya sido la multiprogramación.
    - El problema era el tiempo de espera, la solución a la que se llegó fue dividir la memoria en varias secciones, con un trabajo distinto en cada partición. Mientras un trabajo estaba esperando que terminara su E/S, otro podía estar usando la CPU. Si se podían tener en la memoria principal suficientes trabajos a la vez, la CPU podía mantenerse ocupada casi todo el tiempo. También, tenían la capacidad de leer trabajos de las tarjetas al disco tan pronto como se llevaban al cuarto de computadoras. Luego, cada vez que un trabajo terminaba su ejecución, el sistema operativo podía cargar uno nuevo del disco en la partición que había quedado vacía y ejecutarlo.
- Cuarta generación
  - 1980-actualidad
  - Computadoras personales
    - Con la invención de los circuitos integrados a gran escala (LSI), chips que contienen miles de transistores en un cm<sup>2</sup> de silicio, nació la era de la computadora personal.
    - Dos sistemas operativos dominaron inicialmente el campo de las computadoras personales y las estaciones de trabajo: MS-DOS de Microsoft y UNIX. MS-DOS se usaba ampliamente en la IBM PC y otras máquinas basadas en la CPU Intel 8088 y sus sucesoras. Más tarde, la Pentium y Pentium Pro. Aunque la versión inicial de MS-DOS era relativamente primitiva, versiones subsecuentes han incluido características más avanzadas, muchas de ellas tomadas de UNIX. El sucesor de

Microsoft para MS-DOS, Windows, originalmente se ejecutaba encima de MS-DOS, pero a partir de 1995 se produjo una versión autosuficiente de WINDOWS.

- 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 chips RISC de alto rendimiento.

## Sistemas operativos obsoletos

- Windows phone
- Symbian
- Temple OS
- Minix
- Windows ME
- Windows 95
- Raspberry

## KERNEL Y LLAMADAS AL SISTEMA

- Definición
  - Capa fundamental de un SO, es el encargado de comunicar y administrar (asignar y priorizar) los recursos de la computadora, como la RAM o el uso del procesador
  - Interactúa entre las diferentes aplicaciones, y sus necesidades, con los recursos que posee el dispositivo para ejecutarlos (CPU, memoria, dispositivos)
    - Cuando un programa necesita ejecutarse pide los recursos de hardware que necesite al Kernel del SO
  - Asigna prioridades según las necesidades del SO
  - Es el cerebro del SO
    - No forma parte del hardware, hace parte del SO. Es la traducción directa entre el SO y el hardware.
      - usuario --> SO --> Kernel --> Computadora
    - Se encarga de manejar todos los drivers del hardware que están instalados en el equipo
      - Hace interrupciones (pantalla azul) cuando los drivers no se instalan adecuadamente o se da cuenta que un dispositivo/proceso falla. Esto lo hace para evitar daños en el hardware y en el SO
  - Su objetivo es el de optimizar funcionalidades básicas del SO
- Tipos de estructura de un kernel - RAMAS
  - Monolítico
    - El más veloz ya que se comunica con llamadas al sistema
    - El 70% del kernel no es utilizado
      - Se desperdicia mucho espacio de la memoria porque cuando se carga el kernel completo, se cargan drivers y

diferentes métodos para todo tipo de dispositivos que el SO puede operar

- Si un sistema falla, todo el núcleo falla
- Es un código de muchas líneas que está alojado en un solo espacio de memoria y posee todos los drivers, los servicios y los métodos de administración de recursos
- Linux trabaja con esta rama
- Microkernel
  - Más lento debido a que se comunica con paso de mensajes
  - Más fácil agregar nuevas funcionalidades
  - Requiere más líneas de código
  - Solo posee las instrucciones más básicas de administración en un pequeño espacio de memoria y deja a los diferentes dispositivos su propio manejo
  - Una desventaja es que un microkernel pertenece únicamente a un dispositivo y al ser parte fundamental de un SO hay que diseñar un SO por cada dispositivo con su microkernel, a diferencia del monolítico
- Variantes sobre las RAMAS
  - Kernel híbrido
    - Es un microkernel con más código "no esencial", pero menor al de un monolítico puro
    - Agiliza la velocidad de un microkernel
    - Compatible para gran variedad de dispositivos
  - Nanokernel
    - El código es aún más reducido que en microkernel, pero más difícil crear
    - Todos los servicios se comunican con paso de mensajes
    - Fácil modificación del SO
- Llamadas al sistema
  - Son la manera en la cual un programa solicita una acción o servicio al SO con el que interactúa
    - Esta acción es el punto de enlace entre el modo usuario y el modo privilegiado del SO. Lo que permite a las aplicaciones utilizar recursos de hardware.
- Objetivos de las llamadas al sistema
  - Es la forma en la que los SO diferencian las acciones que pueda realizar un usuario (modo usuario) de las que no, ya que algunas pueden llegar a ser muy dañinas para el SO que solo debe controlarlas él mismo (modo privilegiado)
    - El SO autoriza y administra todas las acciones potencialmente riesgosas
    - Un usuario que modifica aspectos claves del SO puede causar grandes daños en el mismo
- Clasificación de llamadas al sistema



- Todas las llamadas al sistema trabajan como unidades de control para el SO, se establecieron cinco tipos de llamadas al sistema:
  - Gestión de control
    - Supervisa el inicio, creación, detención y finalización de los procesos
  - Gestión de archivos
    - Incluyen la creación, eliminación, apertura, cierre, escritura y lectura de archivos
  - Gestión de dispositivos
    - Administra los recursos disponibles, como ser el almacenamiento
  - Gestión de información
    - Asegura la puntualidad e integridad de la información
  - Comunicación entre procesos
    - Coordina la interacción entre los distintos procesos y aplicaciones