Sistema operativo:

El sistema operativo es el programa fundamental de todos los programas, se encarga de traducir el lenguaje digital en uno que el usuario entienda (a través de una interfaz), controla todos los dispositivos del computador (dispositivos de entrada y salida), gestiona (dirige y administra) todos los procesos, programas y aplicaciones, el sistema operativo se encuentra en modo privilegiado, ejemplos tenemos:

Windows, Mac OS X, Linux, Blackberry, Android, Windows Phone, Apple OS.

Las funciones que realiza el Sistema operativo son:

Desarrolla de programas.

Ejecuta programas.

Accede a dispositivos de entrada y salida.

Accede de forma controlada a los archivos.

Acceso al sistema.

Detecta errores y respuestas.

Mostrar estadísticas.

Las responsabilidades del sistema operativo:

Crear y eliminar procesos.

Suspender y reanudar procesos.

Sincronizar y manejar la comunicación entre procesos.

Asignar y recuperar memoria.

Mantener registro de partes ocupadas de la memoria.

Utilización eficiente.

Responsabilidades del SO para administrar dispositivos de E/S:

Sistema de buffers.

Interfaces para drivers de dispositivo.

Drivers para dispositivos específicos de hardware.

Responsabilidades del SO para administración del almacenamiento:

Manipulación de archivos y directorios.

Mapear archivos en almacenamiento secundario.

<https://www.gcfaprendelibre.org/tecnologia/curso/informatica_basica/sistemas_operativos/1.do>

Módulos del sistema operativo:

El SO se conforma de 4 módulos:

Núcleo o kernel.

Administrador de memoria.

Sistema de entrada/salida.

Administrador de archivos.

<http://so-grupo3.blogspot.com/2007/05/cmo-est-conformado-un-sistema-operativo.html>

Sistemas operativos por lotes:

Son sistemas operativos cuyas aplicaciones requieren muy poca o ninguna interacción entre el usuario y la aplicación en ejecución.

<http://hildaquispecorrea.blogspot.com/2011/06/sistema-operativo-por-lotes.html>

Sistemas operativos de tiempo real:

Sistemas operativos donde los usuarios no tienen importancia, sino los procesos, se utilizan en entornos donde son procesados un gran número de eventos o sucesos.

<http://hildaquispecorrea.blogspot.com/2011/06/sistema-operativo-por-lotes.html>

Sistemas operativos de multiprogramación (o multitarea):

Sistemas operativos que soportan la ejecución de múltiples tareas al mismo tiempo, tiempo compartido en sistemas modernos.

<http://hildaquispecorrea.blogspot.com/2011/06/sistema-operativo-por-lotes.html>

Sistemas operativos de tiempo compartido:

Sistemas operativos que son usados por más de una persona al mismo tiempo.

<http://hildaquispecorrea.blogspot.com/2011/06/sistema-operativo-por-lotes.html>

Flujo de datos alternativos en NTFS (Alternate Data Stream):

Es una característica del sistema de ficheros NTFS que consiste en incluir metainformación en un fichero.

<https://www.securityartwork.es/2015/02/23/alternate-data-stream-ads-flujo-de-datos-alternativos-en-ntfs/>

Metainformación (metadatos):

Información de la información, es una información relacionada con la información principal, que los usuarios no pueden ver en sus pantallas.

<https://www.arkaitzgarro.com/xhtml/capitulo-11.html>

Spooling (Simultaneous Peripherial Operation On Line):

Es transferir datos poniéndolos en un área temporal de trabajo, puedo poner otros programas para procesarlos en otro momento. Ayuda a acelerar E/S a través del disco duro.

<http://www.alegsa.com.ar/Dic/spooling.php>

Pseudoparalelismo:

En un sistema operativo de tiempo compartido, el pseudoparalelismo es la ejecución de cada proceso unos cuantos milisegundos.

<http://html.rincondelvago.com/informatica-basica_1.html>

Planificador de la CPU:

Determina que proceso le toca ejecutarse.

Swapping:

Es mover un proceso o parte de él desde la memoria principal a un dispositivo secundario de almacenamiento.

<http://wiki.inf.utfsm.cl/index.php?title=Swapping_y_esquemas_de_asignaci%C3%B3n>

Memoria virtual:

Uso combinado de la memoria RAM en su computadora y espacio temporero en el disco duro.

<https://mxrola.answers.acer.com/app/answers/detail/a_id/8176/~/%C2%BFqu%C3%A9-es-memoria-virtual%3F>

Kernel (núcleo):

Es la parte central del sistema operativo, funciona como intermediario entre el software y el hardware. Es la primera capa de software donde se encuentran los drivers que controlan todos los componentes del hardware como por ejemplo:

Pantalla, cámara, bluetooth, memoria, USB, Wi-Fi, audio, carga, CPU, etc.

<https://www.androidpit.es/que-es-kernel-para-que-sirve>

Modo dual:

El compartir los recursos del sistema requiere que el SO se asegure que un programa incorrecto no pueda propiciar que otros programas se ejecuten de forma incorrecta.

<https://chsosunal20161910071.wordpress.com/2016/03/08/operacion-en-modo-dual/>

Modo usuario:

Es el nivel de privilegio que se le da a un proceso ejecutado por el usuario.

<http://www.ehowenespanol.com/kernel-info_293714/>

Modo kernel:

Otorga un acceso ilimitado a los recursos de la computadora.

<http://www.ehowenespanol.com/kernel-info_293714/>

Llamadas al sistema:

Son interfaces de programación que sirven para poder invocar los servicios que el sistema operativo nos ofrece.

<http://wiki.inf.utfsm.cl/index.php?title=Llamadas_al_sistema>

Registro base:

La dirección de memoria mínima.

Registro límite:

Tamaño del rango.

Buffer de datos:

Es un espacio de la memoria en un disco o en un instrumento digital reservado para almacenamiento temporal de información digital, mientras que está esperando ser procesada.

Rutinas de programación:

Es una parte de un programa que facilita la ejecución del mismo, permiten que los programas sean más simples.

<https://prezi.com/-nvnng8fxiom/rutina-de-programacion/>

Arquitectura monolítica:

Arquitectura de los primeros SO, constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas, de tal forma que cada una puede llamar a cualquier otra.

Ventajas:

Muy eficiente ya que se producen muy pocos cambios de contexto.

Desventajas:

Difícil de depurar, un error en una función se puede manifestar en otra distinta.

<https://prezi.com/2uraxobilh0t/tipos-de-arquitecturas-de-software/>

Arquitectura cliente-servidor:

Donde el software reparte su carga de cómputo en 2 partes independientes, los proveedores de los recursos (servidores) y los demandantes (clientes), pero sin reparto claro de funciones.

Ventajas:

Centralización del control.

Escalabilidad.

Fácil mantenimiento.

Tecnologías maduras y robustas.

<https://prezi.com/2uraxobilh0t/tipos-de-arquitecturas-de-software/>

Arquitectura de 3 capas:

Es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, por ejemplo, separando la capa de datos de la capa de presentación al usuario.

Las 3 capas son:

Capa de presentación:

Captura y comunica la información al usuario.

Capa de negocio:

En donde residen las funciones que se ejecutan.

Se reciben las peticiones del usuario.

Se procesa la información.

Se envían las respuestas tras el proceso.

Capa de datos:

En donde residen los datos.

Encargada de gestionar los datos.

Definir y almacenar.

Consultar.

Manipular.

Controlar.

Ventajas:

Simplifica la compresión y la organización del desarrollo de sistemas complejos.

Reduce las dependencias de forma que las capas más bajas no son conscientes de las superiores.

Esta separación añade una enorme flexibilidad al diseño de la aplicación.

<https://prezi.com/2uraxobilh0t/tipos-de-arquitecturas-de-software/>

Microkernels:

Es un tipo de kernel de un SO que provee un conjunto de llamadas mínimas al sistema (primitivas) para implementar servicios básicos, como, por ejemplo, espacios de direcciones, comunicación entre procesos y planificación básica.

<https://es.wikipedia.org/wiki/Micron%C3%BAcleo>

Máquinas virtuales:

Es un software que simula una computadora y puede ejecutar programas como si fuese una computadora normal.

<https://es.wikipedia.org/wiki/M%C3%A1quina_virtual>

Interfaces de usuario:

Es el medio con el cual un usuario puede comunicarse con una computadora, un equipo o una máquina.

<https://www.ecured.cu/Interfaz_de_usuario>

Command-line interface:

Es un método que permite a los usuarios dar instrucciones a algún programa informático por medio de una línea de texto simple.

<https://es.wikipedia.org/wiki/Interfaz_de_l%C3%ADnea_de_comandos>

Shell:

Es el programa informático que provee una interfaz de usuario para acceder a los servicios del SO.

<https://es.wikipedia.org/wiki/Shell_(inform%C3%A1tica)>

Ejemplos de llamadas al sistema:

Posix

Fork, exec(ve): Crea un proceso.

Wait(pid): Espera un hijo para terminar.

Exit: Termina el proceso actual.

Open: Abre un archivo y retorna un descriptor.

Close: Cierra un archivo abierto.

Read: Lee de un archivo a un buffer.

Write: Escribe de un buffer a un archivo.

Background:

Se utiliza para nombrar a todos aquellos procesos o rutinas de ejecución que se realizan en segundo plano.

<https://es.wikipedia.org/wiki/Background>

Proceso zombie:

Si un proceso hijo termina y nadie lo ha esperado, se queda ocupando memoria como un proceso zombie.

Para evitar crear procesos zombies, se puede:

Esperar que el hijo termine.

Crear hijos no ligados al padre.

Descriptor de archivos:

Es un entero positivo que debe ser exclusivo en cada trabajo.

<https://www.ibm.com/support/knowledgecenter/es/ssw_i5_54/ifs/rzaaxfildes.htm>

Tubería (pipes):

Consiste en una cadena de procesos conectados de forma tal que la salida de cada elemento de la cadena es la entrada de otro.

<https://es.wikipedia.org/wiki/Tuber%C3%ADa_(inform%C3%A1tica)>

Metas para diseñar un sistema operativo:

Metas de usuario: El sistema operativo debe ser conveniente, fácil aprendizaje, confiable, seguro y rápido.

Metas del sistema: El SO debe de ser de fácil diseño, implementación y mantenimiento, a la vez que flexible, confiable, sin errores y eficiente.

Bloque de control del proceso (PCB):

Es un registro especial donde el SO agrupa toda la información que necesita saber acerca de un proceso particular.

<https://es.wikipedia.org/wiki/Bloque_de_control_del_proceso>

Hilos de ejecución:

También hebra o subproceso es una secuencia de tareas encadenas muy pequeña que puede ser ejecutada por un sistema operativo.

<https://es.wikipedia.org/wiki/Hilo_de_ejecuci%C3%B3n>

Planificador a corto plazo (Planificador de la CPU):

Selecciona que proceso debe ser ejecutado y lo aloja en la CPU.

Planificador a largo plazo (Planificador de trabajos):

Selecciona que procesos deben ser traídos a la cola de procesos listos.

Planificador a medio plazo:

Puede ser añadido si el grado de programación múltiple necesita ser disminuido.

Cambio de contexto:

Consiste en la ejecución de una rutina perteneciente al SO multitarea de una computadora, cuyo propósito es parar la ejecución de un hilo para dar paso a la ejecución de otro distinto.

<https://es.wikipedia.org/wiki/Cambio_de_contexto>

Procesos independientes:

No pueden afectar o ser afectados por la ejecución de otros procesos.

Procesos cooperativos:

Pueden afectar o verse afectados por la ejecución de otros procesos.

Stubs:

Trozo de código usado como sustituto de alguna otra funcionalidad. Puede simular el comportamiento de código existente temporal para un código no desarrollado.

Modelos multihilos:

Modelo de uno a uno:

En este tipo de modelo se asigna un hilo de usuario a un hilo de núcleo, es decir, cada hilo de ejecución es un único proceso con su propio espacio de direcciones y recursos.

Modelo muchos a uno:

Este enfoque asigna múltiples hilos de nivel de usuario a un hilo de nivel de kernel.

<https://prezi.com/z9x65mreg76s/multihilos-y-modelos-de-multihilos/>

Grand Central Dispatch:

<https://es.wikipedia.org/wiki/Grand_Central_Dispatch>

Paginación:

Es una estrategia de organización de la memoria que consiste en dividir la memoria en porciones de igual tamaño, llamadas marcos de página o páginas, identificadas por un número, conocido como número de página.

<http://1984.lsi.us.es/wiki-ssoo/index.php/Paginaci%C3%B3n>