

Sistema Operativo

El software más importante

Importancia de los SOs

El SO **es el software más importante** en un sistema informático, pues es la interfaz entre la máquina y los usuarios y programas. Sus funcionalidades principales son:

- Permite **comunicar** al usuario y a otros programas con el hardware, para que no tengan que lidiar con la complejidad de los dispositivos electrónicos.



Importancia de los SOs

- Administra los recursos hardware, asignando de forma controlada los procesadores, memorias, sistemas de archivos y dispositivos de E/S.



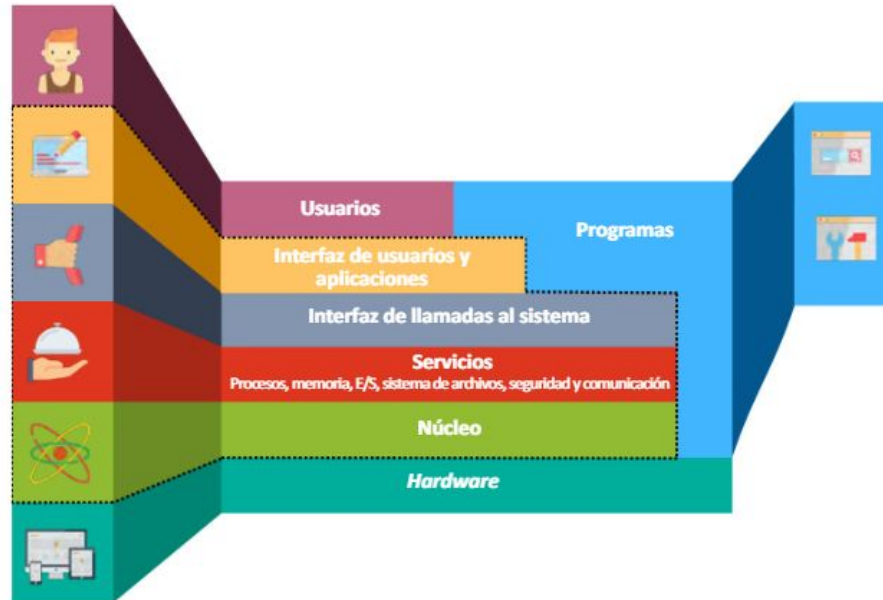
Elementos y estructura

El SO es el software más grande y complejo de un equipo informático, aunque básicamente esté formado por:

- ❑ **El núcleo** (kernel): tiene acceso a todas las operaciones permitidas por el hardware.
- ❑ **Los servicios**: la administración del hardware se realiza bajo la demanda de los servicios, que son solicitados por las aplicaciones a través de una interfaz de llamadas al sistema.
- ❑ **La interfaz de usuario y aplicaciones**: permiten la comunicación entre el usuario y las aplicaciones y el equipo informático.

Elementos y estructura

Aquí puedes observar la estructura y ubicación del SO en un sistema informático.



Procesos

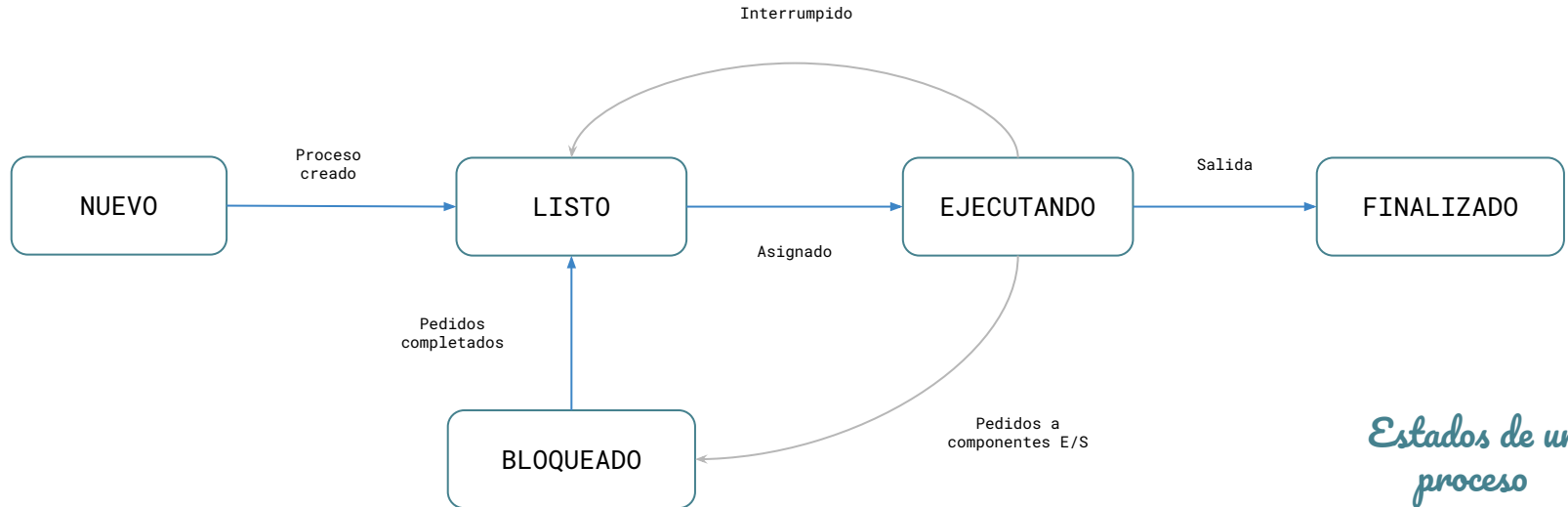
Representan la ejecución de un programa en un entorno de computación. En pocas palabras, podríamos decir que un proceso es un **programa en ejecución en un sistema operativo**. Puede ser una aplicación de usuario, una parte del sistema operativo en sí o incluso un servicio en segundo plano.

Utilizando otras palabras, un proceso es la **instancia activa de un programa** en un momento dado. Incluye el código del programa, sus datos, su estado de ejecución y recursos asignados.

Los procesos tienen acceso a recursos del sistema, como la CPU, la memoria, el almacenamiento y los dispositivos de entrada/salida.

Procesos

Los procesos pueden **comunicarse entre sí** a través de mecanismos de comunicación proporcionados por el sistema operativo. Esto puede incluir la compartición de datos o la sincronización de actividades.



Procesos

Los procesos pueden **crearse y terminarse** dinámicamente. La creación de un proceso generalmente implica **cargar el programa en la memoria** y configurar sus recursos asociados. La terminación del proceso implica **liberar los recursos y eliminarlo del sistema**.

El sistema operativo es **responsable** de administrar la ejecución de múltiples procesos en una CPU. Utiliza **algoritmos de planificación** para determinar qué proceso se ejecutará en un momento dado, con el objetivo de optimizar la utilización de la CPU y garantizar una respuesta eficiente a las solicitudes de los usuarios.

¿Qué es un algoritmo?

En pocas palabras, un algoritmo es un **conjunto preciso y ordenado de instrucciones** o reglas diseñadas para realizar una tarea específica o **resolver un problema** particular.

Los algoritmos son utilizados en diversas disciplinas, incluyendo la informática, las matemáticas, la ingeniería y muchas otras áreas.

Tarea: ¿podrías decirme algún algoritmo que lleves a cabo en el día a día?

A continuación, veremos los algoritmos de planificación más útiles y sencillos. Debemos tener en cuenta que únicamente tenemos un procesador (hay un solo núcleo).

Algoritmos de planificación: FIFO

En el algoritmo de planificación de procesos **FIFO** (First-Come, First-Served), los procesos **se ejecutan en el orden en que llegan a la cola** de listos.

El primero en llegar es el primero en ser atendido. Este algoritmo es simple y fácil de implementar, pero puede llevar a problemas de inanición y no es eficiente en términos de tiempo de espera.

Presta atención al ejemplo del profesor sobre una planificación de procesos usando FIFO.

Algoritmos de planificación: FIFO

Supongamos que tienes una CPU con tres procesos listos para ejecutarse y cada proceso tiene su hora de llegada y su tiempo de ejecución especificados. Usando el algoritmo FIFO, realiza la planificación y terminación de los procesos según la siguiente información:

- Proceso A: Tiempo de llegada = 0, Tiempo de ejecución = 6 unidades de tiempo.
- Proceso B: Tiempo de llegada = 2, Tiempo de ejecución = 4 unidades de tiempo.
- Proceso C: Tiempo de llegada = 4, Tiempo de ejecución = 2 unidades de tiempo.

Considera hacer una gráfica de tiempo para visualizar mejor el ejercicio.

Algoritmos de planificación: SJF

En el algoritmo de planificación **SJF** (Shortest Job First), los procesos **se ejecutan en orden ascendente de duración** estimada.

El proceso más corto se ejecuta primero. Minimiza el tiempo de espera promedio, pero puede causar inanición para procesos largos.

Algoritmos de planificación: SJF

Supongamos que tienes una CPU con cuatro procesos listos para ejecutarse y cada proceso tiene su hora de llegada y su tiempo de ejecución especificados. Usando el algoritmo SJF, realiza la planificación y terminación de los procesos según la siguiente información:

- Proceso A: Tiempo de llegada = 0, Tiempo de ejecución = 6 unidades de tiempo.
- Proceso B: Tiempo de llegada = 2, Tiempo de ejecución = 8 unidades de tiempo.
- Proceso C: Tiempo de llegada = 4, Tiempo de ejecución = 1 unidad de tiempo.
- Proceso D: Tiempo de llegada = 6, Tiempo de ejecución = 9 unidades de tiempo.

Algoritmos de planificación: SRTF

En el algoritmo de planificación **SRTF** (Short Remaining Time First), los procesos **se ejecutan en orden ascendente de duración** estimada, con la particularidad de que cuando un proceso está listo para ser ejecutado, entonces se activa el dispatcher para ver si es más corto que lo que queda por ejecutar del proceso que se encuentra en ejecución (e.d. El SO evalúa cuál es el más corto y le otorga prioridad a dicho proceso).

Algoritmos de planificación: SRTF

Supongamos que tienes una CPU con cuatro procesos listos para ejecutarse y cada proceso tiene su hora de llegada y su tiempo de ejecución especificados. Usando el algoritmo SRTF, realiza la planificación y terminación de los procesos según la siguiente información:

- Proceso A: Tiempo de llegada = 0, Tiempo de ejecución = 6 unidades de tiempo.
- Proceso B: Tiempo de llegada = 2, Tiempo de ejecución = 8 unidades de tiempo.
- Proceso C: Tiempo de llegada = 4, Tiempo de ejecución = 1 unidad de tiempo.
- Proceso D: Tiempo de llegada = 6, Tiempo de ejecución = 9 unidades de tiempo.

Algoritmos de planificación: Round Robin

En el algoritmo de planificación **Round Robin** se asigna un pequeño **intervalo de tiempo** de CPU (quantum) **a cada proceso** en un ciclo circular.

Este algoritmo es justo y evita la inanición, pero puede tener un alto tiempo de respuesta si el quantum es demasiado grande.

Algoritmos de planificación: Round Robin

Supongamos que tienes una CPU con cinco procesos listos para ejecutarse y un quantum de tiempo de 3 unidades. Usando el algoritmo Round Robin, realiza la planificación y terminación de los procesos según la siguiente información:

- Proceso A: Tiempo de llegada = 0, Tiempo de ejecución = 8 unidades de tiempo.
- Proceso B: Tiempo de llegada = 1, Tiempo de ejecución = 6 unidades de tiempo.
- Proceso C: Tiempo de llegada = 2, Tiempo de ejecución = 4 unidades de tiempo.
- Proceso D: Tiempo de llegada = 3, Tiempo de ejecución = 2 unidades de tiempo.
- Proceso E: Tiempo de llegada = 4, Tiempo de ejecución = 10 unidades de tiempo.

Sistema de archivos

Decimos que un sistema de archivos es una **estructura lógica** utilizada por un sistema operativo para organizar, almacenar y gestionar archivos y directorios en dispositivos de almacenamiento, como discos duros, unidades flash, discos ópticos y sistemas de almacenamiento en red.

Proporciona un método estándar para el **almacenamiento** y recuperación de datos, lo que permite a los usuarios y las aplicaciones acceder a archivos de manera organizada y eficiente.

Tarea: busca sistemas de archivos utilizados por diferentes SOs.

Sistema de archivos

Vamos a ver algunas características comunes entre los diferentes tipos de sistema de archivos:

- ❑ **Archivo:** Un archivo es una unidad básica de información almacenada en un sistema de archivos. Puede contener datos de cualquier tipo, como texto, imágenes, programas ejecutables, documentos, etc.
- ❑ **Directorio** (no confundir con “carpeta”): Un directorio es una estructura que se utiliza para organizar archivos en una jerarquía. Los directorios pueden contener archivos y subdirectorios, lo que permite una organización lógica y ordenada de los archivos en el sistema.

Sistema de archivos

- ❑ **Ruta:** Una ruta es una descripción de la ubicación de un archivo o directorio en el sistema de archivos. Puede ser una ruta absoluta (desde la raíz del sistema de archivos) o una ruta relativa (desde un directorio específico).

Tarea: busca y compara una ruta en Windows con una ruta en Linux. ¿Ves la diferencia?

- ❑ **Metadatos:** Los sistemas de archivos almacenan metadatos junto con los archivos, que incluyen información sobre el nombre, tamaño, tipo, fecha de creación, fecha de modificación y permisos de acceso de un archivo o directorio.

Sistema de archivos

- ❑ **Operaciones:** Los sistemas de archivos permiten realizar operaciones comunes, como crear, leer, escribir, borrar, copiar, mover y cambiar el nombre de archivos y directorios.
- ❑ **Integridad y seguridad:** Los sistemas de archivos también pueden implementar mecanismos de seguridad y control de acceso para proteger los archivos y datos contra accesos no autorizados y garantizar la integridad de los datos almacenados.

Sistemas operativos más conocidos

Cada uno de estos sistemas operativos tiene características y diferencias significativas en términos de interfaz de usuario, funcionalidad, compatibilidad de software y hardware, seguridad y otros aspectos.



WINDOWS

7, 8, 10, XP, Vista & Phones



LINUX

Fedora, Redhat
Ubuntu



ANDROID

4.2 & Above



CHROME

OS & Browser

iOS

IOS

6 to 11



MAC OS X

Mavericks, Yosemite
El Captain, Sierra

Tarea: lee los siguientes artículos:

<https://mantenimientoinformaticoeconomico.com/2023/01/21/sistemas-operativos-diferencias-y-similitudes>

<https://www.pccomponentes.com/diferencias-entre-windows-y-linux>