**EXPOSICIÓN DE PROCESOS:**

**Programación distribuida:**

* La programación distribuida consiste en la ejecución de varios procesos concurrentes en un sistema distribuido.
* Se usa también este término para hacer referencia al desarrollo de programas que utilizan varios procesos concurrentes que pueden ejecutarse en distintos procesadores de un sistema distribuido, y a las técnicas que lo hacen posible.
* En un sistema distribuido no hay memoria compartida ni red de conexión específica entre los distintos ordenadores.
* La comunicación entre procesos se realiza mediante mensajes a través de la red de comunicaciones que comunica los distintos ordenadores, para los que se suelen utilizar los protocolos de red estándares TCP o UDP.
* Las principales ventajas de los sistemas distribuidos es que son altamente escalables y configurables.
* Sus principales inconvenientes es que son sistemas heterogéneos y complejos de mantener. La sincronización entre procesos puede ser compilada también. La comunicación entre procesos se realiza a través de la red, que tiene una velocidad limitada, por lo que puede convertirse en un cuello de botella.

**Programación distribuida: ventajas y desventajas:**

* Las principales ventajas de los sistemas distribuidos es que son altamente escalables y configurables.
* Sus principales inconvenientes:
  + Son sistemas heterogéneos y complejos de mantener.
  + La sincronización entre procesos puede ser complicada también.
  + La comunicación entre procesos se realiza a través de la red, que tiene una velocidad limitada, por lo que puede convertirse en un cuello de botella.

**Programación concurrente: ventajas y desventajas:**

* La programación concurrente permite incrementar el rendimiento del sistema.
* También hace posible un menor tiempo de respuesta.
* La mejora en el rendimiento dependerá, de todas formas, del tipo de aplicaciones.
* Lo ideal es que una aplicación se pueda descomponer en procesos independientes que se ejecuten en paralelo con un mínimo de interacción y comunicación entre ellos.
* Un sistema multiprocesador debe ser:
  + Flexible para permitir distribuir dinámicamente la carga de trabajo entre los procesadores disponibles.
  + Escalable para permitir que la potencia de cálculo aumente con la instalación de nuevos procesadores.
* Un sistema multiprocesador redundante tiene procesadores suplementarios que proporcionan:
  + Alta disponibilidad.
  + Tolerancia a fallos
* Los principales inconvenientes de la programación concurrente:
  + Dificultad de implementar mecanismos adecuados de sincronización y comunicación entre procesos.
  + Sobrecarga.
  + La red podría convertirse en un cuello de botella.

**Kernel del SO y llamadas al sistema:**

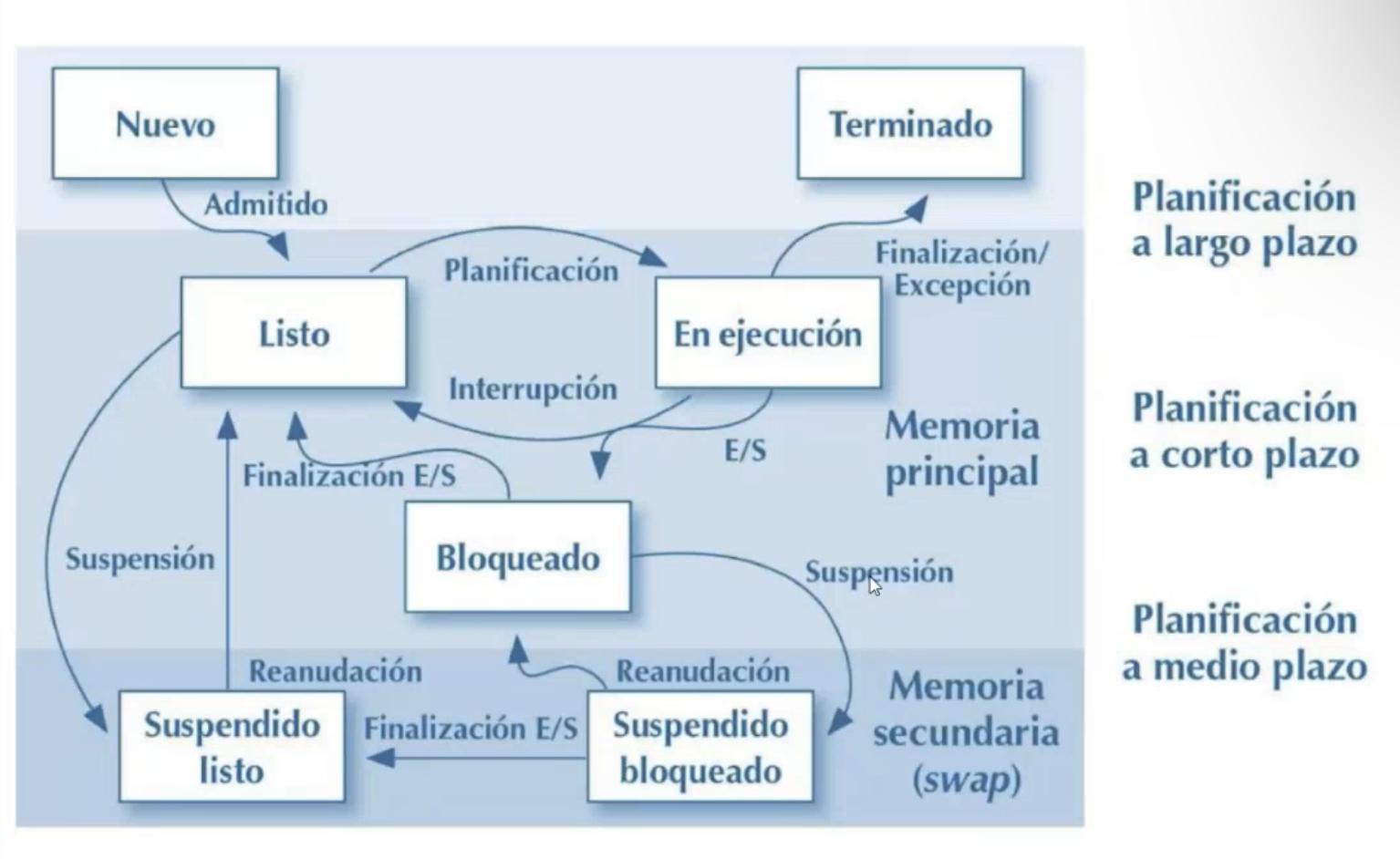
* Es una parte pequeña y muy optimizada del sistema operativo que da respuesta a multitud de eventos mediante un mecanismo de gestion de interrupciones.
* Las interrupciones pueden producirse por diversas causas:
  + Eventos generados por el hardware.
  + Llamadas al sistema. Interrupciones periodicas que incovocan al planificador de procesos a corto plazo del sitema operativo (CPU scheduler).
* Cuando sucede una interrupción el procesador deja de ejecutar el proceso en curso y pasa a ejecutar una rutina de tratamiento de interrupción.
* Cuando finaliza, se reanuda la ejecución en el mismo lugar en que estaba antes antes de la interrupción.
* El procesador tiene dos modos de funcionamiento:
  + Modo kernel, o supervisor, y modo usuario.
  + Las rutinas de tratamiento de interrupciones se ejecutan en modo kernel.

**Linux:**

* Linux no es un sistema operativo, sino un núcleo o kernel que se puso a disposición del público bajo una licencia de software libre GPL.
* En torno a linux se han desarrollado infinidad del público bajo la misma licencia.
* Es más correcto hablar de un sistema operativo GNU/Linux, o de una distribución de GNU/Linux. Este es un sistema operativo construido en torno a un kernel Linux, y con software de sistema desarrollado para el proyecto GNU. Existen unas cuantas distribuciones de GNU/Linux, como Debian, Suse, Ubuntu o MInt.

**Estados de ejecución de un proceso:**

* Desde que se crea inicialmente un proceso, puede pasar por diversos estados y cambiar de uno a otro.
* El sistema operativo gestiona los procesos y realiza los cambios de estado, teniendo en cuenta distintos eventos que pueden suceder durante el ciclo de vida del proceso.
* Los cambios de estado de los procesos están controlados por el sistema operativo, con el objetivo de conseguir:
  + Máximo aprovechamiento de los recursos del sistema, y en particular del procesador o procesadores.
  + Ejecución lo más eficiente posible de todos los procesos.
* Esto se consigue con una planificación a varios niveles.



* Para la gestión de los procesos, el sistema operativo utiliza un bloque de control de procesos (PCB) para cada proceso.
* El sistema operativo mantiene colas de procesos para cada uno de los estados.
* También mantiene colas para cada dispositivo de E/S, con información de los procesos que tienen operaciones pendientes en el dispositivo.
* La utilización de estas colas permite realizar, de manera muy eficiente, la planificación de procesos.

**Hilos y procesos:**

* Para crear un proceso es necesario reservar espacio en la memoria para el. Esto consume una cantidad considerable de tiempo y recursos del sistema.
* Los hilos o hebras (threads en inglés) se llaman a veces procesos ligeros (lightweight processes).
* Un proceso en ejecución tiene inicialmente un hilo, pero se pueden crear más de manera sencilla y rápida.
* La ejecución de un proceso termina cuando finaliza la ejecución de todos sus hilos. Y a la inversa, si se termina la ejecución de un proceso, se acaba también la ejecución de todos sus hilos.
* La creación de un nuevo hilo para un proceso ya existente no requiere reservar e inicializar espacio en la memoria, porque este es compartido por todos los hilos de un mismo proceso.
* Se necesitan mecanismos de sincronización entre hilos para evitar problemas que puedan darse si los distintos hilos modifican sin control objetos situados en la memoria compartida.
* El planificador a corto plazo, o CPU scheduler, gestiona de manera independiente los distintos hilos de un mismo proceso.

**Servicios:**

* Los servicios son un tipo particular de procesos que proporcionan determinados servicios a otros procesos.
* Se ejecutan en segundo plano (en background) y no son directamente utilizados por los usuarios.
* Los servicios se ejecutan continuamente. Normalmente, son iniciados por el sistema operativo durante su arranque, y se ejecutan en segundo plano (en background).
* Suelen proporcionar información acerca de su ejecución en ficheros de log.
* Los servicios pueden proporcionar servicios a otros procesos en el mismo ordenador, y también a otros procesos en otros ordenadores.
  + Suelen comunicarse a través de una red de comunicaciones, utilizando protocolos de red estándares, como TCP o UDP, de la familia de protocolos TCP/IP.
  + Este tipo de servicios en red son un componente fundamental de los sistemas distribuidos.

**Servicios: mejorar tiempo de respuesta:**

* Una técnica frecuente utilizada por los servicios para mejorar el tiempo de respuesta ante peticiones de servicios de otros procesos es crear un nuevo hilo para responder a cada nueva petición recibida, o bien mantener un pool de hilos, de manera que a cada nueva peticion se le puede asignar inmediatamente un hilo ya disponible en el pool.
* De esta forma, el hilo principal queda libre para recibir nuevas peticiones.
* Un servidor operativo tiene sus propios mecanismos para gestionar los servicios. Normalmente, se puede configurar cada uno para que arranque automáticamente cuando arranca el sistema.
* También existen mecanismos generales para arrancar o parar servicios.