

Diferencias Entre Programación Concurrente y Secuencial

• **Programación Secuencial** - Es el enfoque tradicional en el desarrollo de software, donde las instrucciones se ejecutan una tras de otra en un orden específico. Cada instrucción debe completarse antes que la siguiente pueda ejecutarse, aunque es un modelo simple, tiene limitaciones en términos de aprovechamiento de recursos o capacidad de respuesta.

• Características

- **Orden de ejecución**: Las instrucciones siguen un orden fijo y lineal.
- **Simplicidad**: La ejecución lineal simplifica la comprensión y depuración.
- **Predictibilidad**: El flujo de ejecución y los resultados son predecibles.
- **Menor Complejidad**: No implica concurrencia ni sincronización.

• **Programación Concurrente** - Implica la ejecución simultánea de múltiples tareas o procesos en un sistema. Los procesos pueden intercalarse o ejecutarse en paralelo, brindando eficiencia y capacidad de respuesta.

• Características

- **Ejecución Simultánea**: Múltiples tareas pueden ejecutarse al mismo tiempo.
- **Aprovecho de los Recursos**: Se aprovechan de los recursos y se evita la inactividad.
- **Capacidad de Respuesta**: Las aplicaciones pueden responder a entradas mientras se realizan tareas en segundo plano.
- **Complejidad Aumentada**: Se requiere coordinación y sincronización entre procesos.
- **Mayor Rendimiento**: Puede aprovechar el paralelismo en sistemas multicore.

Tipos de Interacción

entre procesos de Programación Concurrente

En la programación concurrente la interacción entre procesos juega un papel crucial en la gestión de la ejecución simultánea de tareas y la coordinación de actividades. Exploraremos los diferentes tipos de interacción entre los procesos Prog. Conc. Detectando como los procesos pueden comunicarse, coordinarse y colaborar de manera eficiente.

Tipos de Interacción:

- **Comunicación Directa:** Implica la transferencia explícita de datos o informaciones de procesos concurrentes. Esto se logra mediante mecanismos como colas, tuberías y memoria compartida. Esta forma de comunicación permite a los procesos compartir información de manera controlada y predecible.

- **Comunicación Indirecta:** Ocurre cuando los procesos interactúan a través de intermediarios o estructuras compartidas. Los mecanismos como los semáforos y los monitores permiten a los procesos controlar el acceso a recursos compartidos y coordinar sus actividades de manera segura y estructurada.

- **Sincronización:** Es esencial en la Prog. Conc. para garantizar el orden y la correcta ejecución de los procesos. Los mecanismos de sincronización, como los semáforos y los bloques, permiten a los procesos coordinar sus acciones y asegurarse de que se cumplan ciertas condiciones, antes de continuar.

- **Coordinación y Colaboración:** Los procesos pueden coordinar sus actividades para lograr objetivos comunes. Esto puede implicar dividir una tarea en subprocesos que trabajan juntos para completarla. La colaboración implica la comunicación y el intercambio de resultados parciales para lograr un objetivo conjunto.

- **Exclusión Mutua:** Es un concepto clave en la Prog. Conc. para evitar conflictos y garantizar que un recurso compartido solo sea accedido por un proceso a la vez. Mecanismos como los semáforos binarios y los bloques aseguran que un solo proceso pueda acceder al recurso en un momento dado.