



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:	21/09/2023
Hora de inicio:	09:00

Introducción a la Programación y Computación 1 [Sección B]

Diego Fernando Cortez Lopez

Agenda

- Anuncio
- Clase 8
- Dudas
- Ejemplo Práctico



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Introducción a NodeJS y Angular

IMPARTIDO POR: **Pedro Ordoñez**



05 de octubre
17:30 – 18:30 HORAS

 Google Meet



/ecysFIUSAC



<https://dtc-ecys.org>

Concurrencia y Paralelismo

Clase 8

Concurrencia y Paralelismo

En la actualidad, la programación y la informática desempeñan un papel fundamental en prácticamente todos los aspectos de nuestra vida cotidiana. Desde aplicaciones móviles y redes sociales hasta sistemas de control de tráfico aéreo y análisis de datos, la informática impulsa el funcionamiento de nuestro mundo cada vez más digitalizado.

Sin embargo, el desafío clave al que se enfrentan los desarrolladores de software y los ingenieros informáticos es la necesidad de hacer que las aplicaciones y sistemas sean más eficientes, rápidos y capaces de manejar una creciente cantidad de datos y tareas. Y aquí es donde entra en juego la gestión de tareas simultáneas, un concepto esencial que subyace en el corazón de la programación moderna y la informática. De aquí surge el concepto de concurrencia y paralelismo, siendo dos conceptos que se relacionan con la ejecución de tareas simultáneas.

Concurrencia

La concurrencia se refiere a la capacidad de un sistema o programa para manejar múltiples tareas aparentemente al mismo tiempo. Aunque en realidad las tareas se ejecutan de forma intercalada y comparten recursos, la concurrencia crea la ilusión de que se están ejecutando simultáneamente.

Características clave de la concurrencia:

1. **División de tareas:** En un entorno concurrente, las tareas se dividen en subprocesos o procesos más pequeños que pueden ejecutarse en paralelo o de manera intercalada.
2. **Multitarea:** La concurrencia permite que múltiples tareas se ejecuten de manera concurrente en un sistema operativo o una aplicación.
3. **Gestión de recursos compartidos:** Los subprocesos o procesos concurrentes a menudo comparten recursos como la memoria, la CPU o los archivos. La gestión de estos recursos compartidos es crucial para evitar problemas como condiciones de carrera.

Paralelismo

El paralelismo se refiere a la ejecución real simultánea de tareas en múltiples unidades de procesamiento independientes. En otras palabras, en un entorno paralelo, las tareas se ejecutan verdaderamente al mismo tiempo en hardware separado.

Características clave del paralelismo:

1. **Múltiples núcleos o procesadores:** El paralelismo se logra utilizando hardware especializado, como múltiples núcleos de CPU o múltiples procesadores.
2. **Aumento de rendimiento:** El principal objetivo del paralelismo es mejorar el rendimiento al ejecutar tareas de manera simultánea y, por lo tanto, reducir el tiempo de ejecución total.

Diferencias clave entre Concurrencia y Paralelismo

Ejecución simultánea real vs. ilusión de simultaneidad:

Concurrencia:

En la concurrencia, las tareas se ejecutan de manera intercalada, compartiendo recursos, pero creando la ilusión de que se están ejecutando simultáneamente. En realidad, el sistema o la aplicación alterna entre las tareas rápidamente.

Paralelismo:

El paralelismo implica la ejecución real simultánea de tareas en múltiples unidades de procesamiento independientes, como núcleos de CPU separados o procesadores. No hay intercalación; las tareas se ejecutan al mismo tiempo en hardware separado.

Diferencias clave entre Concurrencia y Paralelismo

Hardware requerido:

Concurrencia:

La concurrencia no requiere hardware específico para lograr la ilusión de simultaneidad. Puede lograrse en sistemas de un solo núcleo o procesador.

Paralelismo:

El paralelismo requiere hardware específico, como múltiples núcleos de CPU o procesadores, para lograr la ejecución real simultánea de tareas.

Diferencias clave entre Concurrencia y Paralelismo

Objetivo principal:

Concurrencia:

La concurrencia a menudo se utiliza para mejorar la capacidad de respuesta y la gestión de tareas en aplicaciones, permitiendo que las tareas se ejecuten de manera intercalada y evitando bloqueos.

Paralelismo:

El paralelismo se utiliza principalmente para aumentar el rendimiento al ejecutar tareas de manera simultánea y, por lo tanto, reducir el tiempo de ejecución total de un conjunto de tareas.

Diferencias clave entre Concurrencia y Paralelismo

Ejemplos de aplicaciones:

Concurrencia:

Un navegador web que muestra páginas web mientras descarga contenido en segundo plano es un ejemplo de concurrencia. También se utiliza en aplicaciones con un solo hilo de ejecución que deben manejar múltiples tareas concurrentemente.

Paralelismo:

Un servidor web de alto rendimiento que atiende múltiples solicitudes de clientes de manera simultánea utilizando múltiples núcleos de CPU es un ejemplo de paralelismo. También se encuentra en aplicaciones de procesamiento intensivo, como el renderizado de gráficos o la simulación científica.

Diferencias clave entre Concurrencia y Paralelismo

Resultados y beneficios:

Concurrencia:

La concurrencia puede mejorar la capacidad de respuesta y la gestión de tareas en aplicaciones, pero no siempre conduce a un aumento significativo en el rendimiento, especialmente en sistemas con un solo núcleo.

Paralelismo:

El paralelismo tiende a aumentar directamente el rendimiento al aprovechar al máximo el hardware disponible, lo que puede llevar a una ejecución más rápida de tareas.

Procesos



Procesos

Los procesos son un concepto fundamental en la programación y los sistemas operativos. Representan la ejecución de un programa en un entorno aislado, lo que permite la ejecución concurrente de múltiples aplicaciones.

Un proceso es una instancia en ejecución de un programa. Cada proceso tiene su propio espacio de memoria, registro y recursos asignados, lo que lo hace independiente de otros procesos en el sistema.

Procesos

Los sistemas operativos permiten la creación y administración de procesos. Esto incluye la asignación de recursos, la programación de procesos para su ejecución y la terminación de procesos cuando ya no son necesarios.

Los procesos pueden comunicarse entre sí utilizando mecanismos de comunicación interprocesos. Esto permite compartir datos y coordinar tareas entre procesos separados.

Los sistemas operativos utilizan procesos para ejecutar aplicaciones y servicios. Por ejemplo, cada aplicación que ejecutas en tu computadora generalmente se ejecuta como un proceso separado.

Subprocesos



Subprocesos

Los subprocesos, también conocidos como hilos (threads), son unidades más pequeñas de ejecución dentro de un proceso. A diferencia de los procesos, que son independientes y tienen su propio espacio de memoria, los subprocesos comparten el mismo espacio de memoria de su proceso principal. Esto hace que los subprocesos sean más ligeros en términos de recursos y más eficientes para ciertas tareas que requieren paralelismo.

Ventajas de utilizar subprocesos

- **Mayor rendimiento:** Los subprocesos permiten que las aplicaciones realicen múltiples tareas de manera simultánea, lo que puede mejorar significativamente el rendimiento y la velocidad de ejecución.
- **Mejora de la capacidad de respuesta:** Al ejecutar tareas en segundo plano mediante subprocesos, una aplicación puede mantener una interfaz de usuario sensible y receptiva mientras realiza operaciones intensivas en CPU.
- **Aprovechamiento de CPU multinúcleo:** En las CPU modernas, que a menudo tienen múltiples núcleos, los subprocesos pueden aprovechar al máximo los recursos disponibles al realizar tareas en paralelo.
- **Mejora en la escalabilidad:** Las aplicaciones que utilizan subprocesos pueden manejar mejor múltiples solicitudes concurrentes, lo que es esencial para servidores y aplicaciones web de alto tráfico.

Diferencias entre procesos y subprocessos

- **Espacio de memoria:** Los procesos tienen su propio espacio de memoria independiente, mientras que los subprocessos comparten la misma memoria del proceso principal.
- **Creación y terminación:** Los procesos son más costosos en términos de tiempo y recursos para crear y terminar, mientras que los subprocessos son más ligeros y rápidos de crear y destruir.
- **Comunicación:** Los procesos se comunican entre sí a través de mecanismos de comunicación interproceso, como tuberías o mensajes, mientras que los subprocessos pueden compartir variables y datos directamente.

Ejemplos de implementación de subprocesos en aplicaciones

- **Aplicaciones en tiempo real:** En aplicaciones de tiempo real, como videojuegos, los subprocesos se utilizan para gestionar la física del juego, la lógica de la inteligencia artificial y la representación gráfica de manera simultánea.
- **Procesamiento paralelo:** En aplicaciones de procesamiento de datos intensivos, como el procesamiento de imágenes o el análisis de datos, los subprocesos pueden dividir la carga de trabajo para acelerar el proceso.
- **Aplicaciones web:** En servidores web, los subprocesos se utilizan para manejar múltiples solicitudes de clientes de manera concurrente, permitiendo que varios usuarios interactúen con la aplicación al mismo tiempo.

¿Dudas?

Tarea 3

Elaborar un resumen y comentario sobre una conferencia a la que hayas asistido, y adjuntar capturas de pantalla del comienzo y el final de la conferencia. Las conferencias elegidas son las siguientes::

- El papel Fundamental de la Inteligencia Artificial en la Cuarta Revolución Industrial
 - martes 26/09 9:00 p.m.
- Tendencias Tecnológicas para Emprendimientos en Guatemala
 - martes 26/09 2:00 p.m.
- Factores de Éxito o Fracaso en proyectos de Software
 - jueves 28/09 10:30 a.m

Link del cronograma de las conferencias:

https://www.canva.com/design/DAFuY9IEhnE/ch0XSN41m1UL3puvLJOkZA/view?utm_content=DAFuY9IEhnE&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink