

Programación Concurrente

La programación concurrente es un paradigma orientado a la creación de aplicaciones eficientes y flexibles en sistemas modernos, donde que múltiples tareas se ejecutan de manera simultánea, midiendo tanto la capacidad de respuesta como el rendimiento del sistema.

Se abordará la diferencia entre conceptos estrechamente relacionados: Computación concurrente y la Computación paralela. Aunque ambas implican la ejecución de diferentes tareas o procesos, abordan esta ejecución de manera distinta al usar objetivos y recursos específicos.

Proceso en Programación Concurrente

En programación concurrente un proceso es una entidad ejecutable independiente que representa la ejecución de un programa, es una parte de este. Cada proceso tiene su propia espacio de memoria, recursos y contexto de ejecución. Los procesos pueden ejecutarse en paralelo, lo que significa que múltiples procesos pueden estar activos al mismo tiempo y ser ejecutadas por múltiples núcleos del CPU, o hilos.

Cada proceso tiene su propio ciclo de vida, que generalmente es constante de varios estados como: creado, listo, en ejecución, Bloqueado y terminado. La representación de procesos es la razón de cambiar la CPU de un proceso a otros o sistemas operativos intercambiar operaciones de proceso asegurándose que los recursos se asignen adecuadamente y que los procesos se ejecuten de manera justa y eficiente.

Ejemplos de Proceso

Imagina que ejecuta un navegador web y un reproductor de música como procesos separados (S.O). Ambos procesos pueden estar activos al mismo tiempo permitiéndole revisar por internet mientras que escucha música. Cada proceso tiene su propia colección de recursos y se ejecuta en su sólo espacio de memoria lo que sucede que un proceso no tiene acceso a otro.

Circunstancia de Programación Concurrente

La concurrencia se refiere a la capacidad de un sistema para manejar distintas tareas de manera simultánea o en rápida secuencia. En programación concurrente a concurrencia se logra mediante la ejecución de múltiples procesos o hilos. Los procesos o hilos pueden ejecutarse en paralelo (en sistemas con múltiples núcleos) o mediante una ejecución rápida entre ellos (en sistemas de un solo núcleo), lo que crea la ilusión de ejecución simultánea.

La concurrencia permite que múltiples tareas se ejecuten al mismo tiempo, lo que puede aumentar la eficiencia y la eficacia de respuesta del sistema. Sin embargo, también introduce problemas como la gestión de recursos compartidos y la sincronización para evitar problemas como las condicione de Currus y Bloques.

Ejemplo de Concurrencia

Un ejemplo que está desarrollando una aplicación de edición de imágenes permite aplicar filtros y ajustar en tiempo real, mientras que se carga una imagen, pueden aplicar la concurrencia para que la aplicación cargue la imagen de un blob, mientras otro hilo se encarga de aplicar los filtros en paralelo. De este modo, si varía el tamaño, una interfaz más fluida y receptiva ya que la carga y el procesamiento no se bloquean entre sí. Sin embargo, es importante implementar el código de forma cuidadosa para evitar posibles colisiones al usar el blob e intentar acceder a la imagen que trabaja este hilo.

Computación Concurrente

La programación concurrente se refiere a la ejecución de múltiples tareas o procesos en un sistema, donde estos procesos pueden ejecutarse simultáneamente y sincronizar sus resultados diferentes. El objetivo principal de la programación concurrente es mejorar la eficiencia de respuesta, la eficiencia de una aplicación al permitir que diferentes tareas se ejecuten de manera intercalada, aunque estos procesos no se ejecutan necesariamente al mismo tiempo. Se obtiene rápidamente es la respuesta, gracias a la ejecución de una tarea simultánea.

En la computación concurrente, los procesos a menudo interactúan entre sí. Pueden compartir recursos o comunicarse mediante mecanismos de sincronización. Sin embargo, esta interacción puede dar lugar a efectos como condicione de carrera y bloques que requieren una gestión cuidadosa.

Computación Paralela

Se centra en la ejecución simultánea real de múltiples tareas o procesos en diferentes unidades de procesamiento. El objetivo principal radica en acelerar la ejecución de una tarea al dividirla en partes más pequeñas que pueden ejecutarse simultáneamente en diferentes núcleos del CPU o en unidades separadas en una red. La computación paralela requiere de hardware más avanzado, como procesadores multicore o sistemas cluster. En la computación paralela específica, cada tarea se divide en subprocesos o subproblemas independientes, cuyos resultados se combinan para obtener el resultado final.

de procesamiento y rendimiento. Sin embargo, la computación paralela también presenta desafíos como necesidad de coordinar y sincronizar los subprocesos para garantizar resultados correctos y precisos.

Diferencias Claves entre C. Concurrente y C. Paralela

- 1.- Efecto de la computación concurrente se centra en administrar de manera eficiente múltiples tareas interrelacionadas, mientras que la computación paralela se enfoca en la ejecución simultánea real de tareas divididas en subprocesos independientes.
- 2.- Diferencia: la computación concurrente crea una ilusión de ejecución simultánea entre tareas, mientras que la computación paralela implica la ejecución real en el hardware paralelo.
- 3.- Objetivo: la computación concurrente tiene como objetivo mejorar la eficiencia y la capacidad de respuesta del sistema. La computación paralela busca aumentar la velocidad de procesamiento y el rendimiento general de las tareas.
- 4.- Interacción: la computación concurrente implica la interacción entre los diferentes procesos concurrentes, en la computación paralela los subprocesos pueden estar más aislados y requieren menor comunicación.

Características de la Programación Concurrente

- Ejecución Simultánea: la programación concurrente permite la ejecución simultánea a múltiples tareas o procesos. Estos procesos pueden ejecutarse en diferentes núcleos del CPU, o bien, aprovechar el máximo de los recursos disponibles.
- Mínimización de la latencia: Una de las principales ventajas de la PC es mejorar la eficiencia y el rendimiento del sistema. Al ejecutar tareas en paralelo se reduce el tiempo total de la ejecución.
- Capacidad de respuesta: la computación concurrente tiene la capacidad de responder a las aplicaciones al permitir que los procesos se ejecuten de forma interrelacionada. Esto significa que una aplicación puede responder a las interacciones del usuario, mientras se realizan otras tareas en segundo plano.

- Descomposición de Problemas: Mediante la Pearson los problemas se pueden dividir en cuatro más pequeños que se pueden abordar de manera independiente. Cada parte puede ejecutarse en paralelo o bien separada, facilitando la resolución de problemas complejos.
- Convergencia y Divergencia: Los procesos convergen a medida medida convergen y sincronizarse entre sí. Los procesos el convergente, como los colas y los señales, permiten diferentes datos y coordinar actualizaciones.
- Recursos Compartidos: los procesos concurrentes pueden compartir recursos para memoria y archivos. Sin embargo el acceso simultáneo de recursos compartidos puede dar lugar a problemas como condicione de cierre, por lo que se requiere sincronización adecuada.
- Paralelismo vs Distribución: La programación concurrente abarca tanto para sistemas multihilo como la de elaboración de sistemas distribuidos. Ambas abarcan desde aplicaciones en una sola máquina hasta aplicaciones distribuidas en múltiples servidores.
- Manejo de la Concurrencia: la programación concurrente implica lidiar con conflictos específicos de concurrencia como condiciones de cierre, bloques y dead lock. Si se requieren interacciones coordinadas para estos procedimientos.
- Modelos de Programación: Se utilizan diferentes modelos de programación concurrente, como hilos, procesos ligeros, concurrencia basada en eventos y modelo de actores. Cada modelo tiene sus propias ventajas y desventajas.
- Dificultad y Complejidad: la programación concurrente puede ser más compleja que la programación secuencial los errores pueden ser difíciles de depurar y reproducir debido a la naturaleza interrelacionada de las ejecuciones.
- Planificación de Procesos: los gestores operativos desempeñan un papel crucial en la programación concurrente al administrar la planificación y asignación de recursos de los procesos. La gestión de prioridades y los cronogramas de procesos son aspectos fundamentales.