

DOCUMENTO DE DESAFIO TEORICO DECHALLENGE DE MERCADO LIBRE

Por: Oscar Bohórquez

Fecha: 19-Mayo-2022

Procesos, hilos y co-rutinas

- ¿Un problema en el que usaría procesos para resolverlo y por qué?

Un caso de uso posible, sería un sistema de Monitoreo y Control para artefactos hospitalarios (respiradores, bombas diálisis, etc), debido a que estos artefactos son de misión crítica y realizan cantidad de operaciones y cálculos, donde es importante la continuidad, la recuperación operacional y seguridad en el proceso.

- ¿Un problema en el que usaría hilos para resolverlo y por qué?

Un caso podría ser para aquellas aplicaciones, servicios, o portales que tengan que realizar varias tareas de Entrada/Salida como consultas a Base de Datos, lectura de archivos o consumo de recursos externos (rest/soap). Si se utilizan hilos, estos permiten realizar diversas tareas simultáneamente, sin la necesidad de depender del resultado de otras tareas, permitiendo compartir recursos, priorización de tareas y velocidad de respuesta.

- ¿Un problema en el que usaría corrutinas para resolverlo y por qué?

Un caso podría ser la integración con un sistema de Pasarela de Pagos (Ejm: pagos online). Si se utilizan corrutinas se deja al sistema que usa la Pasarela de Pagos en espera (suspendido), mientras que éste responde; esto permite que ambas tareas trabajen cooperativamente y se optimicen recursos (memoria, cpu)

Optimización de Recursos del Sistema Operativo

Si tuvieras 1.000.000 de elementos y tuvieras que consultar para cada uno de ellos información en una API HTTP. ¿Cómo lo harías? Explicar.

Lo haría utilizando programación concurrente (hilos). Donde inicialmente determinaría el número de hilos óptimo para configurar el ThreadPool; seguido, ejecutaría para cada hilo un bloque de peticiones, y finalmente para cada consumo sobre el API lo haría de forma asíncrona.

Análisis de Complejidad

- Dados 4 algoritmos A, B, C y D que cumplen la misma funcionalidad, con complejidades $O(n^2)$, $O(n^3)$, $O(2^n)$ y $O(n \log n)$, respectivamente, ¿Cuál de los algoritmos favorecerías y cuál descartarías en principio? Explicar por qué.

Rta: Se favorece el algoritmo D $\{O(n \log n)\}$, ya que rompen el problema en varios trozos para dar solución al problema; además este tipo de algoritmos se consideran altamente eficiente, ya que la relación entre el número de operaciones y el tamaño de la entrada disminuye y tiende a cero cuando aumenta n . Se descarta el algoritmo C $\{O(2^n)\}$ esto debido a que es de complejidad exponencial y es señal de que el algoritmo necesita una revisión urgente, y es mejor no utilizarlo.

- Asume que dispones de dos bases de datos para utilizar en diferentes problemas a resolver. La primera llamada AlfaDB tiene una complejidad de $O(1)$ en consulta y $O(n^2)$ en escritura. La segunda llamada BetaDB que tiene una complejidad de $O(\log n)$ tanto para consulta, como para escritura. ¿Describe en forma sucinta, qué casos de uso podrías atacar con cada una?

Rta: BetaDB como repositorio principal para aplicaciones de alta transaccionalidad como por ejemplo un banco. Y AlfaDB repositorio para aplicaciones generalmente de consulta, como por ejemplo portales de pymes.