

Auxiliar 2 - CC31A Programación de software sistema

Auxiliar: Paul Leger

27 de abril de 2008

Pregunta 1

Escriba un programa que utilice una función que busque en forma binaria y paralela un elemento en un arreglo desordenado, es decir, cada vez que divida el arreglo ud. debe utilizar threads para buscar ambas mitades . La función debe recibir el arreglo, el elemento inicial y el elemento final.

a) Con este método genera un árbol de threads en paralelo. Ahora limite este árbol paralelo hasta una altura indicada. Desde de ahí debe funcionar de manera secuencial.

b) Ahora cuando un thread encuentre el elemento, los demás hilos no deben seguir buscando el elemento.

c) Ahora contabilice cuantos hilos se crearon para antes de encontrar el elemento buscado.

Pregunta 2

Escriba un programa que simule el problema del productor y consumidor en una cola. Además cuente el número de operaciones hechas por el productor más el consumidor. El control de operaciones de insertar y extraer debe ser controlado por un monitor.

Pregunta 3

Implemente un programa que simule el problema de lectores y escritores en almacén compartido. Varios lectores pueden estar leyendo el almacén compartido, pero cuando un escritor está escribiendo en el almacén no debe haber lectores ni otros escritores accediendo a la almacén compartido.

¿Puede ocurrir que un escritor nunca acceda a la base de datos?

Pregunta 4

Escriba un programa que simule el problema de multi-productores y multi-consumidores en una cola.

¿Cuales son las nuevas condiciones del problema de los productores y consumidores?

Pregunta 5

Escriba un programa que simule el problema del coordinador. El problema del coordinador es cuando hay varios threads que están enviando peticiones a un dispositivo de zona crítica. Para decidir quién tiene derecho al dispositivo el thread coordinador decide y asigna el uso del dispositivo.

Pregunta 6

Implemente el algoritmo de ordenamiento llamado Quicksort de manera concurrente.

Pregunta 7

Implemente un simulador de *PrintJobs* en una impresora. Cada trabajo enviado es un *thread* independiente llamado *worker*. Además hay un *thread* llamado *coordinador* que recibe las peticiones de *PrintJobs*. Los *workers* dejan sus peticiones en una cola. El *coordinador* extrae las peticiones de la cola y procesa la impresión.