

Programación de Servicios y Procesos

Tema 2. Programación Multihilo

Actividad 2.1

1. (C). Escribir una aplicación que cree dos hilos. El primero escribirá en pantalla los números impares del 1 al 99, y el segundo escribirá los pares del 2 al 100. Se crearán dos hilos distintos que no tomarán parámetros ni retornarán ningún valor. El hilo principal deberá esperar a la finalización de estos antes de que la aplicación termine. ¿Qué ocurre si el hilo principal no espera la finalización de los hilos creados? ¿El programa tendría el mismo comportamiento? ¿Qué modificaciones serían necesarias realizar en caso afirmativo?
2. (C). Escribir una aplicación que cree tres hilos, cada uno de ellos recibirá tres argumentos. El primero será un carácter, el segundo será el número de veces a escribir dicho carácter, y el tercer argumento es el tiempo que tardará en escribir cada carácter en segundos. El tiempo está comprendido entre 1 y 6 segundos. Mientras pasa el tiempo, se imprimirá el carácter ".". El hilo principal deberá esperar la finalización de los hilos creados.
3. (C). Escribir una aplicación que cree tantos hilos como filas tenga una matriz de dos dimensiones. Esta matriz se declarará de forma global, con lo que no habrá que pasarla como argumentos a cada hilo. Las dimensiones estarán especificadas en las macros NUM_FILAS y NUM_COLUMNAS. Cada hilo sumará los valores de cada fila y lo retornará al hilo principal, que será el encargado de mostrar los resultados finales. Debido a que el orden de ejecución de los hilos es indeterminado, no sabremos el valor de la suma a que fila corresponde. Desarrollar una solución para este problema, ya que la función principal debe imprimir los resultados de la suma según el orden de las filas.
4. (C). Escribir una aplicación que cree dos hilos. Uno de ellos escribirá la cadena "ping", y otro escribirá la cadena "pong". La aplicación deberá escribir una secuencia alternada de mensajes "pingpong", de forma que, la cadena siempre empiece por "ping", se alternen y no se repitan de forma consecutiva. Ejemplo de salida : *ping pong ping pong ping pong ...*
5. (C). Escribir una aplicación que simule la siguiente situación: Un agricultor tiene una alberca destinada al riego de sus cultivos. La alberca tiene una cantidad de litros inicial, y una cantidad de litros máxima. Cuando riega, el agricultor abre la salida con un caudal establecido fijo, y una engrada de agua del exterior con otro caudal

establecido. El sistema debe simular esta situación, teniendo en cuenta el caudal de agua de entrada, y el caudal de agua de salida y avisar en las dos siguientes situaciones críticas : **1. alberca llena** (evita el desbordamiento de agua) **2. alberca vacía** (avisa al agricultor de la ausencia de agua de riego). La simulación irá informando del estado de la alberca cada segundo.

6. (*Java*). Escribir una versión del ejercicio 4 (ping/pong).
7. (*Java*) Implementa un programa que calcule el factorial de los números que se le pasan como argumentos. El programa creará tantos hilos como argumentos se introduzcan y cada uno de los hilos recibirá uno de esos argumentos y calculará su factorial. No importa que se mezcle la salida que imprimen los hilos.
8. (*Java*) Realizar una aplicación para calcular la suma de los elementos de un vector por partes. El hilo principal deberá crear un número determinado de hilos, y repartir la suma del vector entre todos ellos. Cada hilo deberá recibir como parámetro el índice inicial y final de cada fragmento de array a sumar. Cada hilo acumulará la suma de los elementos del subvector correspondiente sobre una variable accesible desde el hilo principal. Una vez finalizados todos los hilos, la suma final la realizará el hilo principal, que se encargará de sumar las sumas parciales de cada hilo. Probar con un array de 1000 elementos en un rango de 0 a 10, y realizar las sumas parciales mediante 10 hilos.
9. (*Java*) Utilizando un array de números aleatorios de un tamaño determinado, obtener el mayor valor, el menor, la cantidad de números pares, la cantidad de impares, la cantidad de números primos, la media aritmética, el número de veces que aparece un número dado, y la cantidad de números perfectos. Cada una de estas operaciones la realizará un hilo. El hilo principal será el responsable de mostrar la información que recogerá de cada hilo.
10. (*Java*). Crear una clase **Contador** que contendrá un atributo de tipo entero y dos métodos, uno para incrementar y otro para decrementar dicho atributo, además de un constructor que permita inicializar el contador a un valor determinado y un método *get()* para obtener el valor del contador. Usar el objeto **Contador** como un recurso compartido entre dos *Threads*, uno que lo incrementará 1000 veces, y otro que lo decrementará 1000 veces. Mostrar el valor final del objeto compartido. ¿Es el valor esperado? Ejecutar el programa varias veces. ¿Siempre se obtiene el mismo valor? En caso de que el resultado sea incorrecto, realizar las modificaciones necesarias para corregir el funcionamiento.
11. (*Java*). A partir de un objeto **Datos** que contine 2000 números aleatorios cuyos valores están comprendidos entre 0 y 50.000, procesarlos de forma paralela para determinar su paridad, su número de dígitos, si es primo o compuesto, y si es perfecto o no. Se ira mostrando la posición del número, su valor, y sus propiedades.
12. (*Java*). Crear una aplicación para multiplicar matrices usando multihilos. Recordad que multiplicar una matriz consiste en operar una fila de la matriz A con una columna

de la matriz B. Esto determinará el número de hilos a crear. ¿Se necesita usar algún método de sincronización?

**HAPPY
PROGRAMMER
DAY**