Programación multiproceso - Programación multihilo

Aclaración: En este examen se evalúan contenidos específicos de programación concurrente y paralela. Cada ejercicio **debe** ser hecho con las técnicas requeridas, en caso contrario el ejercicio será puntuado con **0 puntos**.

Ejemplo: Escribe un programa que cree un proceso hijo con fork, el hijo escribirá “Hola mundo” y el padre esperará a que finalice el proceso hijo.
Entrega: El alumno entrega un programa que escribe “Hola mundo” (Similar al que se hace el primer día de clase de programación).
Resultado: **0 puntos**.

Penalizaciones:
Cada línea de código con mala sangría o sangría inconsistente restará **0.2 punto**.
Cada número mágico en el código restará **0.2 puntos**. Los valores constante deben ser CONSTANTES (escritas con nombre mayúsculas)

Ejercicios **5 puntos** cada uno.

1. Generador de primos
2. Impresora

Ejercicio 1

Crea un programa que genere un array de enteros con 256 números primos aleatorios entre 0 y 1000

NOTA: son 256 primos aleatorios entre 0 y 1000, NO son los primeros 256 primos.

Estos números aleatorios los generan distintos threads, cada uno tendrá unos índices con los que trabajar.

El programa principal espera a que se generen todos los primos.

El programa tratará de verificar si la suma de cada posición par con el número siguiente es a su vez primo. Esta tarea de verificación está dividida en threads verificadores. Estos threads reciben el array de enteros y un array de booleans (que deberán rellenar), junto con los índices en que que tienen que trabajar.

NOTA: Cada thread verificador sumará la posición par con la siguiente impar, con lo que el número de resultados es 256 / 2, es decir 128 posiciones. El array de booleanos tendrá 128 posiciones.

NOTA: Tanto el número de enteros original como el número de threads que generan y verifican serán potencia de dos. Esto implica que la división del trabajo es entera (no hay posiciones que puedan pertenecer a dos threads)

El programa principal esperará a los threads verificadores.

Por último el programa contará la cantidad de números primos una vez sumados (es decir, contará el número de true en el array de booleanos) y mostrará el % de estos:

n.º primos / total de números * 100

5 Puntos

1 Principal, crea arrays, crea threads y distribuye el trabajo

1 Threads generadores de primos

0.5 Esperar a threads

1,5 Threads verificadores

1 Cálculo del % correcto

Ejercicio 2

Vamos a realizar un programa para la gestión de un servicio de impresión.

Vamos a crear una clase Documento que recibe en su constructor el contenido como una cadena.

Documento doc = new Documento(“Hola mundo”);

El método toString de la clase documento devuelve su contenido.

Vamos a crear una clase Oficinista, esta clase en su constructor recibe el nombre del oficinista y un número de documentos que va a imprimir.

En su método run espera un tiempo aleatorio entre 5 y 10 segundos, e intenta imprimir un documento. Este proceso lo hará n veces, siendo n el número de documentos pasados en su constructor.

El contenido de cada documento es su nombre y doc1, doc2, doc3, etc.

Ejemplo:

Oficinista o = new Oficinista(“Jorge”, 5);

Imprimirá 5 documentos: “Jorge doc1”, “Jorge doc2”, “Jorge doc3”, “Jorge doc4”, “Jorge doc5”

NOTA: esperando el tiempo aleatorio entre cada uno de ellos.

Por último tenemos la **Impresora**.

Esta clase tiene el método imprimir(Documento doc) que cambia su estado a ocupado y escribe el siguiente mensaje

IMPRIMIENDO: <ContenidoDelDocumento>

Esperará 100ms tras imprimir cada letra

También tiene los métodos estáImprimiendo que devuelve true si está imprimiendo y liberarImpresora que cambia su estado a libre

La impresora implementa un patrón singletone para que cada Oficinista pueda acceder a ella.

Ten en cuenta que si la impresora está ocupada el Oficinista deberá esperar a que se libere.

NOTA: Este ejercicio tiene similitudes con el problema de los Raveros.

Puntos 5

0.5 Documento

2 puntos Oficinista con acceso a la impresora como región crítica (Solo uno a la vez)

2,5 punto Impresora

- 0,5 singletone
- 1 imprimir
- 1 gestión del estado