

24 Horas de Le Mans – Programación Concurrente

Para el desarrollo este ejercicio se utilizaron listas y monitores, la metodología implementada a través de estos es muy simple. Dan aviso de cuando un elemento sale de una lista para ingresar en la otra. Dado el contexto, cuando un “Auto” ingresa a Boxes y cuando sale de Boxes e ingresa a la “Pista”.

Es completamente seteable, desde la cantidad de “corredores” hasta los “pits” habilitados.

Sobre el código:

class Autos: Se crean instancias de Autos que son las que van a estar compitiendo por ver quien consigue ingresar a Boxes, caso contrario, seguirán esperando hasta tanto habilite el monitor la salida del mismo para permitir el ingreso de otro “Auto”.

class Pista: mantiene las instancias de Auto un tiempo de espera hasta que se liberen nuevamente para poder intentar ingresar a Boxes. Esto es aleatorio y esperando la señal del monitor.

Ventajas de concurrencia: Se me presento esta idea desde una de las primeras clases, al oír que los “hilos” compiten entre sí, el desarrollo del problema no fue un inconveniente en sí, puesto que la idea la venia pensando desde hace tiempo. El uso de listas me facilitó la implementación del monitor, y por eso me volqué a su utilización. De mas está aclarar que no hay, a mi entender, mejor ejemplo para evidenciar un caso de concurrencia, que una carrera de autos.

Material consultado de la red:

<http://webdiis.unizar.es/~ezpeleta/lib/exe/fetch.php?media=misdatos:pc:10.pdf>

http://www.laminfo.com/blog/archivos/___Teoria_1_Programacion_Concurrente.pdf

El desarrollo del problema y su solución es íntegramente idea de Sebastián Brandariz.

La culpa de que a partir de ahora no deje de buscar y generar nuevos desafíos que impliquen concurrencia, es del Profesor Sebastián Pedersen.