Pablo Eduardo López Ontiveros 25/11/2019

1. Contesta las siguientes preguntas:
2. Describa con sus propias palabras el concepto de “future”:

R= Es algo que representa alguna operación o alguna función que lo vamos a completar en el futuro.

1. Describa brevemente que es exclusión mutua (mutual exclusión):

R= Se usan en la programación concurrente para evitar que algunos códigos o secciones con errores se metan de lleno a los recursos que no deben porque ser compartidos.

1. Defina a que se refiere el concepto de condición de carrera (race condition):

R= Se refiere a que cuando un sistema o algo así intente realizar muchas operaciones al mismo tiempo, pero tienen una cierta naturaleza y esas operaciones se deben hacer en orden en una secuencia correcta y si no, se presenta esta condición de carrera.

1. Mencione al menos tres nombres de los algoritmos de Dekker y describa brevemente la idea central de los mismos:

R=

1.- Alternancia estricta: Se le llama así porque obliga a que cada proceso tenga un turno, significa que hay un cambio de turno cada vez que un proceso sale de una sección critica.

2.-Problema de interbloqueo: Se debe a que si en cada ráfaga del CPU, cada proceso se va a quedar en su mismo estado, y pues garantiza la exclusión mutua pero no la espera limitada.

3.- Postergación indefinida: Se debe a que en una parte de nuestro código (entre la línea 12 y 30) se le coloca un retardo aleatorio de tiempo y puede ser tan grande que no se sabe cuando entra en la sección critica.

1. Describa cuales son las funciones de “Task Parallel Library (TPL):

R= Son tipos públicos y APIS que son parte de los namespaces y se usan para trabajar con los procesos, tiene distintas funciones como las siguientes:

1.- Crear e iniciar las tareas.

2.- Cancelar tareas.

3.- Esperar a que pase el tiempo para pasar tareas.

4.- El manejo de las excepciones.

1. Explique brevemente la diferencia entre ejecución en paralelo y ejecución concurrente

R= El paralelo se usa cuando tiene que ejecutar varios programas se tiene que sobreponer al tiempo usando muchos procesadores y el concurrente se reserva esa energía, igual se va a ejecutar el programa, pero no se tiene que sobreponer al tiempo.

1. Describa a que se le conoce como interbloqueo:

R= Cuando dos procesos llaman al bloqueo juntos. Las dos banderas se quedan prendidas y los dos procesos se esperan a que se liberen los recursos.

1. Explique el concepto de “spinning”

R= Es un gasto de procesador por parte del CPU por lo que se debe utilizar cuando esperamos que la condición de desbloqueo se reciba pronto, sin embargo, impide la latencia y el sobre trabajo.

1. Explique lo que es atomicidad, sus características más relevantes y de un ejemplo

R= Son operaciones atómicas que no son interrumpidas, las que no son atómicas son vulnerables a la race condition, un ejemplo sería el “x = 2” es una operación atómica porque ya esta asignada en la misma operación.

1. Explique lo que es la sección crítica y porque es importante

R= Tiene que ver con los procesos concurrentes que no se pueden ejecutar de manera concurrente y no se puede dividir, es importante ya que si un proceso entra a ejecutar una sección critica, entonces otro proceso no puede ejecutar otra sección critica hasta que sepamos el error.

1. Describa para que sirven los semáforos en programación concurrente

R= Esta relacionado con el tema de los hilos y con los recursos compartidos, para resolver esos problemas se usan los semáforos, funcionan en base en cooperar entre los procesos y se usan las señales para que un proceso se detenga cuando le llegue la señal.

1. Describa a que se refiere la sincronización en programación concurrente

R= Es lo que nos permite coordinar las acciones de los hilos para que obtengamos un resultado predecible.

1. Describa las características de la programación reactiva

R= Es otra forma de concurrencia, donde nuestro programa ya reacciona a los eventos.

Pero se construye sobre eventos asíncronos en lugar de operaciones asíncronas.

1. Describa las características más importantes de SemaphoreSlim

R= Es una clase de C#, que nos permite crear semáforos ligeros, que nos permiten limitar los procesos que pueden acceder a los recursos.

Se les llaman semáforos ligeros a los que están en C# y que no son delegados como tal delegados del sistema operativo que los usa en este caso Windows.

1. En base a las siguientes cadenas, realice un algoritmo que determine cuando una de las cadenas incorrecta y la posición del error. Suba el código a su repositorio con el nombre PV2doParcial y escriba en el examen el SHA correspondiente al commit
2. () () ( ( () ) X
3. ( () ( ( () ) ) X
4. ( ( ( () () ) ) ) X
5. ( () () ) ( ( ) ) <-

Sha= [85c941e](https://github.com/rockfuerte/UNEDL2019B/commit/85c941eaf64bda3e831a37d8ff157d7aaf409425)