1. **Describa con sus propias palabras el concepto de “future”**

**R=** Es como un dato que realizara una operación mas adelante y se completara

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

**R=** Es el control de cuantos threads pueden trabajar en una misma región de un codio

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

**R=** Es cuando en un sistema o dispositivo se quiere realizar muchas operaciones al mismo tiempo pero por su naturaleza este no puede con la tarea.

1. **Mencione al menos tres nombres de los algoritmos Dekker y describa bevemente la idea central del mismo**

**R=** Es un algoritmo para la programación concurrente el cual realiza operaciones de exclusión mutua, que permite a dos procesos o threads compartir un recurso sin dificultades técnicas. 1.- Alternancia estricta: obliga a cada proceso a tener un turno de ejecución. 2.- Problema de Interbloqeo: Cuando dos procesos quieren acceder a la misma tarea, si el proceso entra ambos quedaran atascados y así evitar la exclusión mutua 3.- Colisión región critica no grantisza la exclusión mutua: Cuando dos o mas procesos intentan acceder a la región critica por culpa de como se establecieron los permisos

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

**R=**  Son un conjunto de APIS publicas que forman parte de los namepsaces que se usan para trabajar con threads

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

**R=** La diferencia recae en que la paralela es usado en sistemas donde se ejecutan varios programas que se superponen en el tiempo mediante el uso de varios procesadores y la concurrente reserva para un potencial paralelismo, en el cual la ejecución podría no superponerse en el tiempo.

1. **Describa a que se le conoce como interbloqueo**

**R=**  Cuando dos procesos hacen llamada al bloqueo al mismo tiempo

1. **Explique el concepto de “spinning”**

**R=**  Es en el que un thread espera a que una condicion de desbloqueo que gira entorno a un ciclo de revisión.

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

**R=**  una operacion es atómica cuando no puede ser interrumpida, estas son vulnerables a las condiciones de carrera. ejemplo: z=1 ya que la asignación se hace en una sola operacion

1. **Explique lo que es la sección critica y porque es importante**

**R=** es como una parte de los procesos concurrentes que no pueden ejecutarse de manera concurrente o que su composicion es atómica y no pueden dividirse. Es importante por que si un proceso entra a ejecutar una sección critica en la que se accede a variables compartidas, eso quiere decir que otro proceso no puede entrar a ejecutar una sección critica en la que esas variables se modifiquen en el interior.

1. **Describa para que sirven los semáforos en programación concurrente**

**R=** Es como el encargado de determinar que proceso puede hacer una tarea, sin uno los procesos estarían estancándose en las tareas ya ocupadas, por ejemplo un hilo puede estar modificando mientras otro también haciendo lo mismo.

1. **Describa a que se refiere la sincronización en programacion concurrente**

**R=** permite la coordinación de acciones de los threads para obtener un resultado ya visto

1. **Describa las características de la programación reactiva**

**R=**  Esta relacionada con la programación asíncrona pero construye eventos asíncronos en lugar de operaciones asíncronas, un evento asíncrono podría no tener un inicio, podría pasar en cualquier momento y podría presentarse muchas veces.

1. **Describa las características mas importantes de SemaphoreSlim**

**R=**  nos permite crear semáforos ligeros, que nos permiten limitar el numero de threads que pueden acceder a un recurso o un conjunto de recursos

1. **En base a las siguientes cadenas, realice un algoritmo que determine cuando una de las cadenas es 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 a.(() () (( ()) X b.**