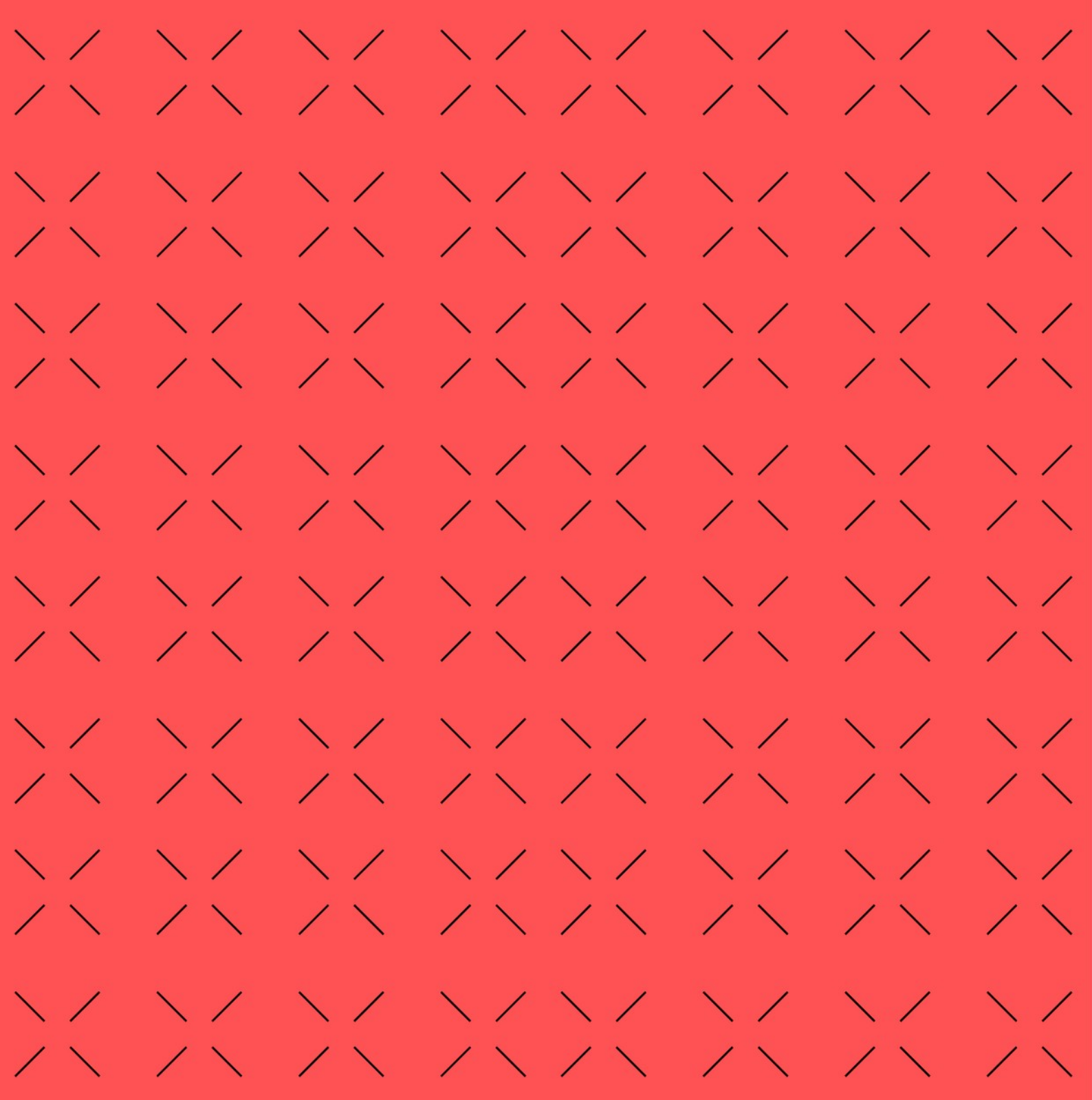


Unidad 5.1

Programación asíncrona



Índice

1	Introducción.....	4
2	Referencias.....	7

Licencia



Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):

No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

1 Introducción

La programación asíncrona ha emergido como una práctica estándar en el desarrollo de aplicaciones programas y sitios web en un entorno informático cada vez más complejo. Con el avance de la tecnología y la presencia de procesadores multinúcleo, la programación asíncrona se ha vuelto esencial para mejorar la eficiencia y la capacidad de respuesta en el desarrollo de software. En entornos cliente-servidor la programación asíncrona responde a la necesidad de manejar operaciones que pueden llevar tiempo, como solicitudes a servidores, descargas de recursos y procesamiento de datos, de manera eficiente sin bloquear la ejecución principal de una aplicación.

¿Qué es la programación asíncrona?

La programación asíncrona es una metodología que se centra en la ejecución de tareas de forma no bloqueante e independiente a la aplicación principal, permitiendo que estas puedan ejecutar operaciones como lectura/escritura de archivos, acceso a bases de datos, o solicitudes a servidores de forma paralela. Una vez acaban las tareas notifican a la aplicación o hilo principal de ejecución que han completado su tarea o si ha ocurrido un fallo.

Por contra, en la programación síncrona el programa esperar pasivamente a que una tarea se complete antes de pasar a la siguiente.

Programación Síncrona:

- En la programación síncrona, las operaciones se ejecutan una tras otra en un orden secuencial.
- Cada operación debe completarse antes de que la siguiente pueda comenzar.
- Puede haber bloqueos, donde el programa espera a que una operación, como la lectura de un archivo o una solicitud de red, se complete antes de continuar con la siguiente tarea.

Programación Asíncrona:

- En la programación asíncrona, las operaciones no bloquean el hilo principal de ejecución, permitiendo que otras operaciones se realicen mientras una espera su finalización.
- Se utilizan mecanismos como callbacks, promesas o `async/await` para manejar la lógica asíncrona de manera más estructurada.
- Es eficiente en situaciones donde hay operaciones de entrada/salida (E/S) que podrían consumir tiempo, ya que permite que otras operaciones se realicen durante la espera.

Ventajas de la Programación Asíncrona:

- **Reducción de Tiempos de Espera:**
La programación asíncrona reduce o elimina los tiempos de espera, mejorando así la eficiencia y la velocidad de ejecución.
- **Mejora de la Capacidad de Respuesta:**
Permite que una aplicación responda a eventos y solicitudes mientras realiza tareas en segundo plano, mejorando la experiencia del usuario.
- **Eficiencia en Operaciones de E/S:**
Es especialmente eficaz para operaciones de entrada/salida (E/S) como lectura/escritura de archivos y solicitudes de red, ya que no bloquea el hilo principal.
- **Manejo Concurrente de Tareas:**
Facilita la ejecución simultánea de tareas independientes, mejorando la eficiencia del procesamiento.

Cuándo Utilizar la Programación Asíncrona:

- **Operaciones de E/S intensivas:**
Situaciones en las que se leen o escriben datos en disco, se realizan operaciones de red o se interactúa con bases de datos pueden ser intensivas en E/S y beneficiosas para la programación asíncrona.
- **Solicitudes de red:**
Al realizar solicitudes a servidores remotos a través de la red, la programación asíncrona permite que otras tareas continúen ejecutándose mientras se espera la respuesta del servidor.
- **Interactividad en interfaces de usuario:**
En aplicaciones de interfaz de usuario (UI), la programación asíncrona es esencial para mantener la capacidad de respuesta. Permite que la interfaz de usuario responda a eventos mientras se realizan otras operaciones en segundo plano.
- **Procesamiento paralelo:**
Cuando hay operaciones independientes que pueden ejecutarse simultáneamente sin depender una de la otra, la programación asíncrona permite aprovechar el paralelismo de manera más eficiente.
- **Manejo de múltiples conexiones simultáneas:**
En servidores que deben manejar múltiples conexiones de clientes simultáneamente, la programación asíncrona puede ser útil para gestionar eficientemente estas conexiones sin necesidad de un thread por conexión.

- **Escalabilidad:**

En aplicaciones que necesitan manejar un gran número de solicitudes simultáneas, la programación asíncrona puede ayudar a mejorar la escalabilidad y la eficiencia del sistema.

- **Eventos y notificaciones:**

Cuando se trabaja con eventos y notificaciones, la programación asíncrona puede facilitar la gestión de estos eventos sin bloquear la ejecución del programa.

Ejemplos en los que podemos usar la programación asíncrona:

- **Llamadas a una API:**

Imagina que estás desarrollando una aplicación que hace consultas a una API para obtener información en tiempo real. Con programación asíncrona, puedes enviar una solicitud a la API y continuar ejecutando otras partes de tu aplicación sin esperar a que la respuesta de la API llegue de inmediato. Una vez que la API responde, puedes manejar los datos obtenidos.

- **Operaciones de E/S - Descarga de un Servicio Web:**

Supongamos que estás desarrollando un programa que descarga archivos desde un servicio web. En lugar de bloquear la ejecución mientras esperas que los archivos se descarguen, puedes utilizar la programación asíncrona para realizar otras tareas mientras la descarga está en curso.

- **Cálculos de Daños en un Juego:**

En el desarrollo de juegos, especialmente en situaciones donde se realizan cálculos intensivos, como determinar el daño en un combate, la programación asíncrona puede ser valiosa. Puedes realizar cálculos de daños en segundo plano sin afectar la jugabilidad principal.

2 Referencias

<https://learn.microsoft.com/es-es/dotnet/csharp/asynchronous-programming/async-scenarios>

<https://historiadelaempresa.com/programacion-asincrona>

<https://bravedeveloper.com/2022/02/25/que-es-la-programacion-asincrona-y-por-que-deberias-utilizarla/>