

Los patrones de P.C de Modelo de Actores

Es en el que el sistema se organiza en "actores", que son entidades independientes y encapsuladas que realizan cálculos, envían y reciben mensajes, y modifican su propio estado de manera autónoma. En lugar de compartir memoria o datos directamente, los actores se comunican entre sí mediante el envío de mensajes. Este enfoque se utiliza principalmente en sistemas más sencillos para manejar la concurrencia.

Principios del Modelo de Actores

- **Encapsulación de Estado:** Cada actor tiene su propio estado privado y es el único que puede modificarlo. No hay variables compartidas.
- **Comunicación a través de Mensajes:** Se comunican enviándose mensajes asíncronos y cada mensaje es un conjunto de datos que puede incluir las instrucciones.
- **Comportamiento Autocentrado:** Los actores pueden:
 - Crear otros actores.
 - Enviar mensaje a otros actores.
 - Cambiar su comportamiento al recibir un mensaje.

Funcionamiento del Modelo de Actores

- 1.- **Creación del Actor** - Pueden ser creados y eliminados en tiempos de ejecución.
- 2.- **Cola de Mensajes** - Cada uno tiene una cola de mensajes que recibe.
- 3.- **Procesamiento del Mensaje** - Se analiza el contenido y puede realizar varias acciones.
- 4.- **Aislamiento** - No hay memoria compartida entre los actores.
- 5.- **Escalabilidad** - Son fáciles de escalar y distribuir en múltiples procesadores o máquinas.

Ventajas del Modelo de Actores

- 1.- Simplicidad de Concurrencia
- 2.- Escalabilidad
- 3.- Tolerancia a fallos
- 4.- Modularidad

Desventajas del Modelo de Actores

- 1.- Complejidad de Depuración
- 2.- Sobrecarga en la Mensajería
- 3.- Limitaciones de Estados Complejos

Ejemplo: Simulación del Chat

Un sistema de chat donde cada usuario es un actor a qui puedes:

- Enviar mensajes a otros usuarios.
- Recibir mensajes en su cola.
- Crear o cambiar de "Sala" de chat.

En el modelo de actores:

- Los usuarios no comparten datos, solo intercambian mensajes.
- La "Sala" de chat puede ser un actor, que recibe y reenvía mensajes a los usuarios.
- Si un usuario envía un mensaje, la sala actúa como intermediaria y distribuye el mensaje a los usuarios que están en la sala.

Uso en la práctica

En lenguajes como Erlang y Scala (con su librería Akka), que están diseñados con el modelo de actores en mente. También puede implementarse en otros lenguajes, como Python, usando bibliotecas como pyka o implementación personalizada de colas y manejo de mensajes.