

JAVA MULTI-THREAD

DEADLOCK

Palabra clave sincronizada se utiliza para hacer que la clase o método thread-safe, lo que significa que solo un subproceso puede tener bloqueo de método sincronizado y usarlo, otros subprocesos tienen que esperar hasta que las versiones de bloqueo y cualquiera de ellos adquiere ese bloqueo.

Es importante utilizar si nuestro programa se ejecuta en un entorno multiproceso donde dos o más subprocesos se ejecutan simultáneamente. Pero a veces también causa un problema que se llama Deadlock.

Podemos evitar el estado de bloqueo de muertos conociendo sus posibilidades. Es un proceso muy complejo y no es fácil de atrapar. Pero aún así si lo intentamos, podemos evitar esto. Hay algunos métodos por los cuales podemos evitar esta condición. No podemos eliminar por completo su posibilidad, pero podemos reducirla.

Evitar bloqueos anidados. Esta es la razón principal de la cerradura muerta. El bloqueo de muerte ocurre principalmente cuando damos bloqueos a varios hilos. Evite dar bloqueo a varios hilos si ya hemos dado a uno.

Evitar bloqueos innecesarios: Deberíamos haber cerrado solo a los miembros que son necesarios. Tener bloqueo innecesariamente puede conducir a la cerradura muerta.

Uso de la unión de rosas: La condición de bloqueo muerto aparece cuando un subproceso está esperando a otro para finalizar. Si se produce esta condición, podemos usar Thread.join con el tiempo máximo que cree que tardará la ejecución.

LIFECYCLE

Un subproceso en Java en cualquier momento existe en cualquiera de los siguientes estados. Un subproceso se encuentra solo en uno de los estados mostrados en cualquier instante.

- 1.-Nuevo
- 2.-Ejecutable
- 3.-Bloqueado
- 4.-Esperando
- 5.-Timed Waiting
- 6.-Terminado

Nuevo hilo: Cuando se crea un nuevo subproceso, se encuentra en el nuevo estado. El subproceso aún no ha comenzado a ejecutarse cuando el subproceso está en este estado. Cuando un subproceso se encuentra en el nuevo estado, su código aún no se ha ejecutado y no ha comenzado a ejecutarse.

tado de ejecución: Un subproceso que está listo para ejecutarse se mueve al estado de ejecución. En este estado, un subproceso podría estar ejecutándose o estar listo para ejecutarse en cualquier momento. Es responsabilidad del programador de subprocesos dar al subproceso, tiempo para ejecutarse. Un programa multiproceso asigna una cantidad fija de tiempo a cada subproceso individual.

Estado bloqueado/en espera: Cuando un subproceso está temporalmente inactivo, se encuentra en uno de los siguientes estados: Bloqueado Esperando

Espera cronometrada: Un subproceso se encuentra en estado de espera cronometrada cuando llama a un método con un parámetro de tiempo de espera. Un subproceso se encuentra en este estado hasta que se completa el tiempo de espera o hasta que se recibe una notificación.

Estado terminado: Un subproceso finaliza debido a cualquiera de las siguientes razones. Porque existe normalmente. Esto sucede cuando el código de subproceso ha ejecutado completamente por el programa. Porque se ha producido algún evento erróneo inusual, como un error de segmentación o una excepción no controlada.

Ciclo de vida de un hilo