

Interbloqueo e Inanición

Interbloqueo

Situación en que dos o más procesos están esperando por recursos que tiene ocupado el otro. **(2-7)**.

Condiciones necesarias para que se pueda producir interbloqueo:

- Exclusion mutua: existen recursos no compartibles.
- Hold and wait: se permite que un proceso tenga recursos reservados mientras espera para reservar otros
- No apropiación: El sistema operativo no puede liberar recursos reservados.
- Espera circular: Los procesos esperan por recursos reservados por procesos que esperan por los primeros.

Negar cualquiera de estas condiciones lo evita.

Interbloqueo: Tratamiento

El interbloqueo se puede tratar mediante varias técnicas, algunas aplicadas por el SO, otras por el propio sistema concurrente. Algunos SO (como Unix o Windows) ignoran el problema. Es tarea del programador de espacio de usuario evitar que se produzca.

Interbloqueo: Detección

Se permite que ocurran interbloqueos, y se corrigen después. Como el SO conoce que recursos tiene reservados para un proceso, y porque recursos espera puede saber cuando se produce un interbloqueo. Una vez detectado puede

- Matar a los procesos involucrados
- Apropiar recursos reservados y dárselos a otro proceso para resolver la situación

Interbloqueo: Prevención

En prevención se trata de evitar que una de las 4 condiciones necesarias se produzca:

- Exclusión mutua, impidiendo que los procesos tengan acceso exclusivo a los recursos.
- Evitar hold y wait: Haciendo que los procesos reserven todos los recursos necesarios de una vez en vez de mantener reservados unos mientras esperan o Haciendo que los procesos liberen los recursos reservados si no pueden reservar más. **(12)**
- Evitar la no apropiación. Es necesario implementar algún mecanismo de roll-back.
- La espera circular se puede evitar con una reserva ordenada de recursos. **(13,15)**

Interbloqueo: Evitación

-Se intenta prevenir que el sistema entre en una situación de interbloqueo conociendo el estado del sistema y los recursos que los procesos pueden reservar en el futuro. Requiere conocer a priori el uso de recursos que va a hacer un proceso, lo que limita mucho su uso.

Inanición

Un proceso puede estar esperando acceso a un recurso compartido sin conseguirlo. Esta situación se denomina inanición. Por ejemplo, si requerimos la reserva de todos los recursos simultáneamente, los procesos que requieran una gran cantidad de recursos pueden quedar en este estado si hay muchos procesos que requieren pocos compitiendo por ellos. **(17,18)**

Inanición: Preguntas

¿Qué ocurre si hay muchos threads ejecutando bloqueam1, bloqueam2 y bloqueam3? -> Es muy probable que en todo momento alguno de los mutex esté bloqueado.

El thread que ejecuta bloquea mucho no es capaz de entrar en su sección crítica => Inanición