

# Introducción

---

Un interbloqueo es una situación en la que dos o más procesos **quedan bloqueados indefinidamente** al entrar en **conflicto sus diferentes necesidades**.

Ocurren porque **los procesos compiten** por el uso de recursos y porque **se comunican** y sincronizan entre ellos.

Puede ocurrir en muchas situaciones.

## Recursos

Los recursos son **Objetos** que otorga el Sistema Operativo (dispositivo de hardware, un registro de una base de datos...) otorga a los procesos.

### Recursos Apropiativos

Es aquel recurso que se puede quitar al proceso que lo posee sin efectos dañinos (Ej.: la RAM a un proceso).

### Recursos Apropiativos

Es aquel que no se puede quitar a su propietario actual sin hacer que el cómputo falle. (Ej.: cuando un proceso ha empezado a quemar un CD-ROM y le quitamos de manera repentina el grabador de CD; ya que ahora el CD estará lleno de basura).

En general, los recursos que producen interbloqueos son los **No Apropiativos**. Para utilizar un recurso es necesario seguir una serie de pasos:

1. Solicitar el Recurso.
2. Usar el Recurso.
3. Liberar el Recurso.

## Adquisición de Recursos

Una manera de permitir que los usuarios administren los recursos es asociar un semáforo con cada recurso.

Cuando dos procesos compiten por más de un recurso, se adquieren secuencialmente como se muestra en el código siguiente:

```
typedef int    semaforo;
semaforo      recurso_1;
semaforo      recurso_2;

void proceso_A (void)
{
    down (&recurso_1);
    down (&recurso_2);
    usar_ambos_recursos();
    up (&recurso_2);
}
```

```
    up (&recurso_1);
}

void proceso_b (void)
{
    down (&recurso_1);
    down (&recurso_2);
    usar_ambos_recursos();
    up (&recurso_2);
    up (&recurso_1);
}
```

Hay que recalcar que en el anterior código es importante que los semáforos se "upéen" y "downéen" en el mismo orden los 2:

Pongamos la situación de que el Proceso B tiene los 'up' en el orden inverso (primero recurso\_2 y luego recurso\_1): Imaginemos que el Proceso\_A adquiere el down del recurso\_1 y justo en ese momento el Proceso\_B obtiene el quantum y adquiere el down del recurso\_2.

**En este momento el Proceso\_A se queda esperando a que se libere el recurso\_2 y el Proceso\_B se queda esperando a que se libere el recurso\_1 y NO SE AVANZA = INTERBLOQUEO**

## Interbloqueos

Un interbloqueo es, formalmente... *Un conjunto de procesos se encuentra en un interbloqueo si cada proceso está esperando un evento que sólo puede ser ocasionado por otro proceso del conjunto.* Ninguno de los procesos se puede ejecutar, ninguno de ellos puede liberar recursos y ninguno puede ser despertado.

Para este modelo suponemos que los procesos tienen un sólo hilo y que no hay interrupciones posibles para despertar a un proceso bloqueado.

### Condiciones para los Interbloqueos de Recursos