



**Nombre del alumno: Johanne Joaquín Arriaga Díaz.**

**Nombre del profesor: Jonathan Gabriel Hernández.**

**Materia: Sistemas operativos de red.**

**Nombre del trabajo: Mapa conceptual de unidad 3.**

**Grado: Quinto cuatrimestre**

**Grupo: ISC13SDC0119-F**

# Interbloqueo (DeadLock)

## ¿Qué es DeadLock?

Si un conjunto de procesos está en estado de espera por recursos y nunca cambia porque los recursos por los que espera están siendo utilizados por otros procesos en estado de espera.

### ¿Qué debe suceder para que se dé un DeadLock?

**EXCLUSION MUTUA:** Por lo menos un proceso que tenga otorgado un recurso en forma exclusiva.

**USO Y ESPERA:** Al menos un proceso que esté haciendo uso de un recurso y que esté esperando por otros recursos asignados a otros procesos.

**NO INTERRUPCION:** Si un proceso hace uso de un recurso no le podrá ser retirado hasta que voluntariamente el proceso lo libere.

**ESPERA CIRCULAR:** Debe existir un conjunto de procesos esperando por un recurso utilizado por otro en un bucle.

## Postergación indefinida.

En los sistemas que mantienen procesos en espera mientras realizan la asignación de recursos, un proceso puede ser postergado de manera indefinida mientras otro recibe la atención del sistema.

## Mecanismos para tratar con Deadlocks

Si se tiene cuidado al en la forma de asignar los recursos se pueden evitar situaciones de Deadlocks.

### Estado Seguro

Un estado es seguro si se pueden asignar recursos a cada proceso (hasta su máximo) en algún orden sin que se genere DeadLock.

### Algoritmo del banquero de Dijkstra

Asigna peticiones de recursos siempre que den como resultado estados seguros. Solicitudes que den como resultado estados inseguros serán negadas hasta que puedan ser satisfechas evita situaciones de DeadLock asignando los recursos en forma correcta.

Consisten en proveer mecanismos para evitar que se presente una o varias de las cuatro condiciones necesarias del DeadLock.

**Asignar recursos en orden lineal:** Todos los recursos están etiquetados con un valor diferente y los procesos solo pueden hacer peticiones de recursos 'hacia adelante'. Con esto se evita la condición de ocupar y esperar un recurso.

**Asignar todo o nada:** El proceso pida todos los recursos que va a necesitar de una vez y el sistema se los da solamente si puede dárselos todos, si no, no le da nada y lo bloquea.

**Algoritmo del banquero:** Este algoritmo usa una tabla de recursos para saber cuántos recursos tiene de todo tipo.

## Conceptos sobre recursos.

**Recursos no apropiativos:** Un recurso que no se puede liberar antes de completar su actividad sin perder la validez del proceso que lo usa.

**Recursos apropiativos:** Un recurso que puede ser usado temporalmente por varios procesos sin comprometer el correcto funcionamiento de dichos procesos se dice que es un recurso apropiativo.

### Modelo del sistema.

Un sistema se compone de un número finito de recursos que se distribuyen entre varios procesos que compiten por ellos.

### Archivos Secuenciales

Se refiere al procesamiento de los registros, no importa el orden en que se haga, para eso los registros están organizados en forma de una lista y recuperarlos y procesarlos uno por uno de principio a fin.

## Áreas importantes en la investigación del DeadLock

**Recuperación:** Para romper los DeadLock en un sistema, para poder liberarlo de ellos y los procesos estancados pueden continuar y llegar a su fin

**Evitación:** Poner condiciones menos estrictas que la prevención, para lograr una mejor utilización de los recursos, pero procurando no caer en un DeadLock.

**Detección:** Métodos que permiten la ocurrencia de un DeadLock de forma voluntaria o involuntaria.

**Prevención:** Condiciona un sistema para que elimine toda probabilidad de que ocurra un DeadLock.

### Dead lock en sistemas de spool.

Suelen ser los más propensos al dead lock. Varios trabajos generan líneas de impresión a un archivo de spool pueden interbloquearse si el espacio disponible para trabajar se llena antes de completarse alguno de esos trabajos. La solución es limitar los spoolers de entrada de modo que no se lean más archivos cuando se llega al límite de su capacidad.