



## Interbloqueos (Deadlocks)

### Objetivos

- Estudiar el concepto de interbloqueo y los tópicos relacionados a los mismos.
- Reconocer los posibles estados de espera que pueden presentarse entre un conjunto de procesos y un conjunto de recursos.
- Estudiar las técnicas y condiciones bajo las cuales se pueden afirmar o no la existencia de un interbloqueo.
- Emplear los mecanismos y teóricos para detectar y prevenir una situación de interbloqueo en un ambiente simulado.

### Pre-Laboratorio

1. Defina Interbloqueo.
2. Describa las condiciones suficientes y necesarias para que se pueda presentar un interbloqueo en un sistema
3. En el contexto de Interbloqueos: ¿Qué es un estado seguro, inseguro e interbloqueado? ¿Cuáles son sus diferencias?
4. Familiarícese con la representación de estados de interbloqueo investigando la utilidad y notación de un grafo de asignación de recursos y un grafo de espera.
5. ¿Qué consideraciones deben tomarse en cuenta para detectar, evadir y recuperarse de un interbloqueo? Investigue acerca de los algoritmos asociados y las técnicas empleadas para lograr tal propósito tomando en cuenta sus ventajas y desventajas asociadas.
6. ¿La existencia de una espera circular por parte un grupo de procesos implica la existencia de un interbloqueo entre ellos? Justifique su respuesta
7. Usted debe entender y leer el documento anexo con el nombre:  
`user_guide_deadlock.pdf`

### Laboratorio

Discuta con el docente, todos y cada uno de los siguientes pasos además ejecútelos.

1. Verificar la versión de Java

```
java -version
```

2. Verifique la versión de Java que tiene instalada

```
javac -version
```

3. Verifique el valor de la variable de entorno CLASSPATH y asegúrese que incluya el directorio actual en el mismo



4. Cree una estructura de directorio como la siguiente
  - a. Usuario
    - i. Moss
      1. Deadlock
5. Copie el directorio deadlock los archivos empaquetados en deadlock.zip
6. Discuta con el docente la jerarquía de archivos y la funcionalidad de cada uno de los archivos presentados.
7. Ejecute:  

```
java deadlock a 2 1 > a.log
```
8. Intercambie opiniones sobre el proceso de ejecución, así como también los resultados esperados y los resultados obtenidos

## Evaluación

Ejecute las siguientes simulaciones:

1. 

```
java deadlock a 2 2 > simulacion1.log
```
2. 

```
java deadlock b 2 1 1 > simulacion2.log
```
3. 

```
java deadlock b 2 1 2 > simulacion3.log
```

Para cada una de estas simulaciones usted debe explicar el resultado de la simulación, además de generar el grafo de asignación de recursos asociado.

A continuación se desea que usted simule la situación planteada y genere el grafo de asignación de recursos:

En un servidor de impresión un administrador de sistemas ha detectado un baja sustancial en los tiempos de respuestas de las solicitudes a los clientes. Se sospecha de la existencia de un posible interbloqueo entre los procesos que atienden las solicitudes. Sin embargo la existencia de un posible interbloqueo debe corroborarse a fin de evitar confundir la disminución de los tiempos de respuesta debido una alta demanda del servicio durante un período de tiempo. Para ello debe evaluarse el siguiente escenario:

Existen cuatro (4) procesos involucrados [P0, P1, P2, P3] y cinco (5) recursos [R0, R1, R2, R3, R4] con una disponibilidad de 2, 2, 1, 1, 1 instancias respectivamente. Adicionalmente el administrador determinó lo siguiente:

1. P0 se ejecuta 10 unidades de tiempo y necesita una instancia del recurso R1 para ejecutarse 15 unidades de tiempo adicionales, luego solicita una instancia del



recurso R0 para ejecutarse otras 10 unidades de tiempo. Finalmente libera los recursos que tenía asignados y culmina su ejecución.

2. P1 requiere una instancia del recurso R4 pero antes se ejecuta 20 unidades de tiempo. Luego emplea 5 unidades de tiempo en realizar cálculos. Completado éstos cálculos solicita una instancia del recurso 3 y se ejecuta 5 unidades de tiempo adicionales. Finalmente libera los recursos que tenía asignados y completa su ejecución.
3. P2 inicia sus cómputos ejecutándose 10 unidades de tiempo, luego solicita el recurso R0 antes de realizar operaciones por 10 unidades de tiempo adicionales. Para completar sus cálculos solicita una instancia del recurso R2 y procesa datos durante 5 unidades de tiempo. A continuación solicita una instancia del recurso R1 y ejecuta cálculos durante 5 unidades de tiempo adicionales. Para concluir P2 libera los recursos reclamados durante su ejecución.
4. P3 solicita una instancia del recurso R3 después de estar ejecutándose durante 5 unidades de tiempo. Con una instancia del recurso R3, P3 sólo puede procesar datos durante diez unidades de tiempo antes de tener que reclamar una instancia del recurso R2. Seguido de esto P3 se ejecuta durante 15 unidades de tiempo adicionales, libera la instancia R2 que tenía retenida y solicita una instancia de R4. Luego P3 libera la instancia del recurso R3 y la instancia del recurso R4 que tenía asignado.

Valiéndose de la herramienta MOSS simule la situación planteada y genere los archivos respectivos que describen las instrucciones que ejecutan cada proceso `p0.dat`, `p1.dat`, `p2.dat`, `p3.dat`. Asimismo a partir de la traza de salida que usted debe llamar `pout.log` elabore el grafo de asignación de recursos asociado y determine la existencia o no de interbloqueo junto con los procesos y recursos involucrados en el mismo.

#### **Simulaciones:**

**1. 2 puntos**

**2. 3 puntos**

**3. 3 puntos**

**Ejercicio Servidor de Impresión 12 puntos**

**GDSO**