

Sistemas Operativos

Caracterización de interbloqueos

Departamento de Ingeniería en Sistemas y Computación
Universidad Católica del Norte, Antofagasta.

- Un escenario donde pueden aparecer interbloqueos se caracteriza por un conjunto de entidades activas que utilizan un conjunto de recursos para llevar a cabo su tarea.

Partes involucradas:

- Entidades activas: procesos existentes en el sistema
- Recursos del sistema: fisicos (procesador, memoria) o lógicos (archivos, semáforos)

- Un escenario donde pueden aparecer interbloqueos se caracteriza por un conjunto de entidades activas que utilizan un conjunto de recursos para llevar a cabo su tarea.

Partes involucradas:

- Entidades activas: procesos existentes en el sistema
- Recursos del sistema: físicos (procesador, memoria) o lógicos (archivos, semáforos)

- Un escenario donde pueden aparecer interbloqueos se caracteriza por un conjunto de entidades activas que utilizan un conjunto de recursos para llevar a cabo su tarea.

Partes involucradas:

- Entidades activas: procesos existentes en el sistema
- Recursos del sistema: fisicos (procesador, memoria) o lógicos (archivos, semáforos)

- **Reutilizables o consumibles:** El recurso sigue existiendo después de consumirse?
- **Uso dedicado o compartido:** Puede ser usado por varios procesos simultáneamente
- **Con uno o múltiples ejemplares:** Existen varios ejemplares del recurso?
- **Expropiables o no expropiables:** Es posible quitar el recurso cuando se está usando?

- **Reutilizables o consumibles:** El recurso sigue existiendo después de consumirse?
- **Uso dedicado o compartido:** Puede ser usado por varios procesos simultáneamente
- **Con uno o múltiples ejemplares:** Existen varios ejemplares del recurso?
- **Expropiables o no expropiables:** Es posible quitar el recurso cuando se está usando?

- **Reutilizables o consumibles:** El recurso sigue existiendo después de consumirse?
- **Uso dedicado o compartido:** Puede ser usado por varios procesos simultáneamente
- **Con uno o múltiples ejemplares:** Existen varios ejemplares del recurso?
- **Expropiables o no expropiables:** Es posible quitar el recurso cuando se está usando?

- **Reutilizables o consumibles:** El recurso sigue existiendo después de consumirse?
- **Uso dedicado o compartido:** Puede ser usado por varios procesos simultáneamente
- **Con uno o múltiples ejemplares:** Existen varios ejemplares del recurso?
- **Expropiables o no expropiables:** Es posible quitar el recurso cuando se está usando?

- Su vida no depende de su utilización
- Lo son todos los recursos físicos, y algunos lógicos (archivos)

Primer ejemplo de interbloqueo

<u>Proceso P₁</u>	<u>Proceso P₂</u>
Solicita(C)	Solicita(I)
Solicita(I)	Solicita(C)
Uso de rec.	Uso de rec.
Libera(I)	Libera(C)
Libera(C)	Libera(I)

Ejecución con interbloqueo

P₁: solicita(C)
P₂: solicita(I)
P₂: solicita(C) → bloqueo
P₁: solicita(I) → interbloqueo

Ejecución sin interbloqueo

P₁: solicita(C)
P₁: solicita(I)
P₂: solicita(I) → bloqueo
P₁: libera(I)
P₂: solicita(C) → bloqueo
P₁: libera(C)
P₂: libera(C)
P₂: libera(I)

Recursos reutilizables

- Su vida no depende de su utilización
- Lo son todos los recursos físicos, y algunos lógicos (archivos)

Primer ejemplo de interbloqueo

<u>Proceso P₁</u>	<u>Proceso P₂</u>
Solicita(C)	Solicita(I)
Solicita(I)	Solicita(C)
Uso de rec.	Uso de rec.
Libera(I)	Libera(C)
Libera(C)	Libera(I)

Ejecución con interbloqueo

P₁: solicita(C)
P₂: solicita(I)
P₂: solicita(C) → bloqueo
P₁: solicita(I) → interbloqueo

Ejecución sin interbloqueo

P₁: solicita(C)
P₁: solicita(I)
P₂: solicita(I) → bloqueo
P₁: libera(I)
P₂: solicita(C) → bloqueo
P₁: libera(C)
P₂: libera(C)
P₂: libera(I)

- Dejan de existir una vez que un proceso los usa
- Un proceso genera un recurso y otro lo consume
- Asociados a la comunicación y sincronización (mensajes, señales, semáforos)
- Ejemplo de interbloqueo inevitable:

Proceso P₁	Proceso P₂	Proceso P₃
Enviar(P ₃)	Recibir(P ₁)	Recibir(P ₂)
Recibir(P ₃)	Enviar(P ₃)	Enviar(P ₁)
Enviar(P ₂)		Recibir(P ₁)

- Dejan de existir una vez que un proceso los usa
- Un proceso genera un recurso y otro lo consume
- Asociados a la comunicación y sincronización (mensajes, señales, semáforos)
- Ejemplo de interbloqueo inevitable:

Proceso P₁	Proceso P₂	Proceso P₃
Enviar(P ₃)	Recibir(P ₁)	Recibir(P ₂)
Recibir(P ₃)	Enviar(P ₃)	Enviar(P ₁)
Enviar(P ₂)		Recibir(P ₁)

- Dejan de existir una vez que un proceso los usa
- Un proceso genera un recurso y otro lo consume
- Asociados a la comunicación y sincronización (mensajes, señales, semáforos)
- Ejemplo de interbloqueo inevitable:

Proceso P₁	Proceso P₂	Proceso P₃
Enviar(P ₃)	Recibir(P ₁)	Recibir(P ₂)
Recibir(P ₃)	Enviar(P ₃)	Enviar(P ₁)
Enviar(P ₂)		Recibir(P ₁)

- Dejan de existir una vez que un proceso los usa
- Un proceso genera un recurso y otro lo consume
- Asociados a la comunicación y sincronización (mensajes, señales, semáforos)
- Ejemplo de interbloqueo inevitable:

Proceso P₁	Proceso P₂	Proceso P₃
Enviar(P ₃)	Recibir(P ₁)	Recibir(P ₂)
Recibir(P ₃)	Enviar(P ₃)	Enviar(P ₁)
Enviar(P ₂)		Recibir(P ₁)

- En general, los procesos usan ambos tipos de recursos
- Ejemplo de interbloqueo utilizando ambos tipos de recursos:

Proceso P_1	Proceso P_2
Solicita(C)	Solicita(C)
Enviar(P_2)	Recibir(P_1)
Libera(C)	Libera(C)

Si P_2 obtiene C \rightarrow interbloqueo

- En general, los procesos usan ambos tipos de recursos
- Ejemplo de interbloqueo utilizando ambos tipos de recursos:

Proceso P_1	Proceso P_2
Solicita(C)	Solicita(C)
Enviar(P_2)	Recibir(P_1)
Libera(C)	Libera(C)

Si P_2 obtiene C \rightarrow interbloqueo

- Los recursos compartidos los pueden usar varios procesos simultáneamente, por tanto no afectan a los interbloqueos
- Pueden existir recursos con ambos tipos de uso: en la solicitud del recurso debe indicarse el modo de uso deseado
 - Si compartido: concedido si no se está usando en modo exclusivo
 - Si exclusivo: concedido si no se está usando

- Los recursos compartidos los pueden usar varios procesos simultáneamente, por tanto no afectan a los interbloqueos
- Pueden existir recursos con ambos tipos de uso: en la solicitud del recurso debe indicarse el modo de uso deseado
 - Si compartido: concedido si no se está usando en modo exclusivo
 - Si exclusivo: concedido si no se está usando

- Los recursos compartidos los pueden usar varios procesos simultáneamente, por tanto no afectan a los interbloqueos
- Pueden existir recursos con ambos tipos de uso: en la solicitud del recurso debe indicarse el modo de uso deseado
 - Si compartido: concedido si no se está usando en modo exclusivo
 - Si exclusivo: concedido si no se está usando

- Modelo general: N unidades de cada recurso
 - Solicitud de varias unidades de un recurso
 - Ejemplos: sistemas con varias impresoras, memoria, etc.
- Ejemplo: Memoria con 450KB disponibles

Proceso P_1

Solicita(100K)

Solicita(100K)

Solicita(100K)

Proceso P_2

Solicita(200K)

Solicita(100K)

Si P_1 satisface 2 primeras y P_2 satisface 1ª \rightarrow interbloqueo

- Modelo general: N unidades de cada recurso
 - Solicitud de varias unidades de un recurso
 - Ejemplos: sistemas con varias impresoras, memoria, etc.
- Ejemplo: Memoria con 450KB disponibles

Proceso P_1

Solicita(100K)

Solicita(100K)

Solicita(100K)

Proceso P_2

Solicita(200K)

Solicita(100K)

Si P_1 satisface 2 primeras y P_2 satisface 1ª \rightarrow interbloqueo

- Algunas soluciones basadas en expropiación: salvar el estado del recurso y asignarlo a otro proceso.
- Ejemplos de recursos expropiables:
 - Procesador: cambio de proceso
 - Memoria virtual: Reemplazo
- Ejemplo de interbloqueo:
 - P1 espera por el procesador y tiene asignado lector DVD
 - P2 tiene asignado el procesador y solicita lector DVD

- Algunas soluciones basadas en expropiación: salvar el estado del recurso y asignarlo a otro proceso.
- Ejemplos de recursos expropiables:
 - Procesador: cambio de proceso
 - Memoria virtual: Reemplazo
- Ejemplo de interbloqueo:
 - P1 espera por el procesador y tiene asignado lector DVD
 - P2 tiene asignado el procesador y solicita lector DVD

Recursos expropiables o no expropiables

- Algunas soluciones basadas en expropiación: salvar el estado del recurso y asignarlo a otro proceso.
- Ejemplos de recursos expropiables:
 - Procesador: cambio de proceso
 - Memoria virtual: Reemplazo
- Ejemplo de interbloqueo:
 - P1 espera por el procesador y tiene asignado lector DVD
 - P2 tiene asignado el procesador y solicita lector DVD

- Algunas soluciones basadas en expropiación: salvar el estado del recurso y asignarlo a otro proceso.
- Ejemplos de recursos expropiables:
 - Procesador: cambio de proceso
 - Memoria virtual: Reemplazo
- Ejemplo de interbloqueo:
 - P1 espera por el procesador y tiene asignado lector DVD
 - P2 tiene asignado el procesador y solicita lector DVD

Primitivas genéricas

- Solicitud ($R_1[U_1], \dots, R_n[U_n]$)
 - U_1 unidades del recurso 1, etc
 - Si todos disponibles, se concederá
 - Sino se bloquea, pero sin reservar ningún recurso
- Liberación ($R_1[U_1], \dots, R_n[U_n]$)
 - Puede causar desbloqueo de otros procesos

Primitivas genéricas

- Solicitud ($R_1[U_1], \dots, R_n[U_n]$)
 - U_1 unidades del recurso 1, etc
 - Si todos disponibles, se concederá
 - Sino se bloquea, pero sin reservar ningún recurso
- Liberación ($R_1[U_1], \dots, R_n[U_n]$)
 - Puede causar desbloqueo de otros procesos

Representación mediante grafo de asignación

- Nodos $\{N\}$: Procesos $\{P\}$ + Recursos $\{R\}$, asociado a cada R_i va un valor que indica el número de unidades existentes
- Aristas $\{A\}$:
 - Asignación ($R_i \rightarrow P_j$) : P_j tiene asignada una unidad de R_i .
Unidades asignadas no puede exceder unidades existentes
 - Solicitud ($P_j \rightarrow R_i$): P_j tiene pedida y no concedida una unidad de R_i
- Liberación de recursos: Eliminar aristas correspondientes

Representación mediante grafo de asignación

- Nodos $\{N\}$: Procesos $\{P\}$ + Recursos $\{R\}$, asociado a cada R_i va un valor que indica el número de unidades existentes
- Aristas $\{A\}$:
 - Asignación ($R_i \rightarrow P_j$) : P_j tiene asignada una unidad de R_i .
Unidades asignadas no puede exceder unidades existentes
 - Solicitud ($P_j \rightarrow R_i$): P_j tiene pedida y no concedida una unidad de R_i
- Liberación de recursos: Eliminar aristas correspondientes

Representación mediante grafo de asignación

- Nodos $\{N\}$: Procesos $\{P\}$ + Recursos $\{R\}$, asociado a cada R_i va un valor que indica el número de unidades existentes
- Aristas $\{A\}$:
 - Asignación ($R_i \rightarrow P_j$) : P_j tiene asignada una unidad de R_i . Unidades asignadas no puede exceder unidades existentes
 - Solicitud ($P_j \rightarrow R_i$): P_j tiene pedida y no concedida una unidad de R_i
- Liberación de recursos: Eliminar aristas correspondientes

Representación mediante grafo de asignación

- Nodos $\{N\}$: Procesos $\{P\}$ + Recursos $\{R\}$, asociado a cada R_i va un valor que indica el número de unidades existentes
- Aristas $\{A\}$:
 - Asignación ($R_i \rightarrow P_j$) : P_j tiene asignada una unidad de R_i .
Unidades asignadas no puede exceder unidades existentes
 - Solicitud ($P_j \rightarrow R_i$): P_j tiene pedida y no concedida una unidad de R_i
- Liberación de recursos: Eliminar aristas correspondientes

Modelado del interbloqueo

Ejemplo de representación con un grafo y múltiples unidades por recurso

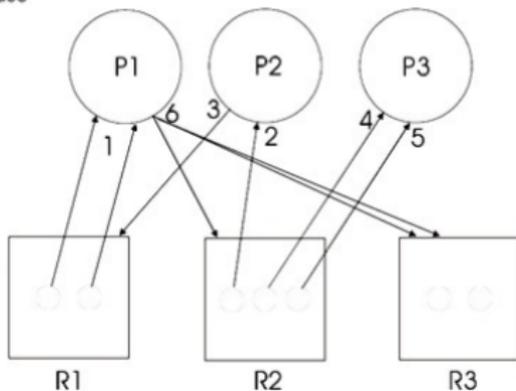
Ejecución de 3 procesos con 3 recursos R_1 (2), R_2 (3) y R_3 (2)

1. P_1 : solicita($R_1[2]$) → solicita 2 unidades
2. P_2 : solicita($R_2[1]$)
3. P_2 : solicita($R_1[1]$) → se bloquea
4. P_3 : solicita($R_2[1]$)
5. P_3 : solicita($R_2[1]$)
6. P_1 : solicita($R_2[1]$, $R_3[2]$) → se bloquea

Grafo resultante:

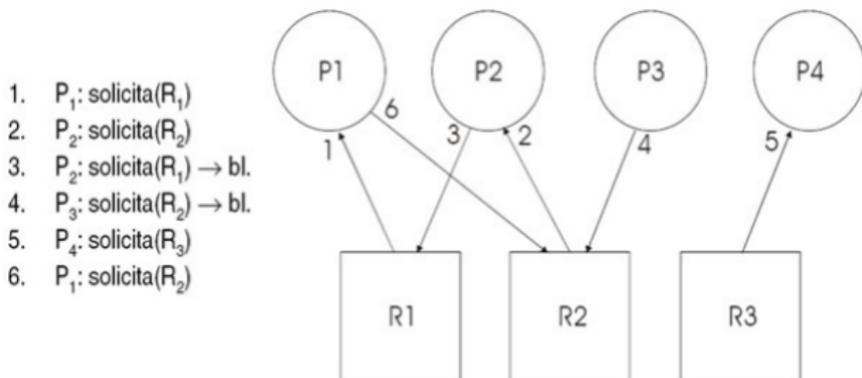
$N = \{P_1, P_2, P_3, R_1(2), R_2(3), R_3(2)\}$

$A = \{R_1 \rightarrow P_1, R_1 \rightarrow P_1, R_2 \rightarrow P_2, P_2 \rightarrow R_1, R_2 \rightarrow P_3, R_2 \rightarrow P_3, P_1 \rightarrow R_2, P_1 \rightarrow R_3, P_1 \rightarrow R_3\}$



Modelado del interbloqueo

Ejemplo de representación con un grafo y una unidad por recurso



$N = \{P_1, P_2, P_3, P_4, R_1(1), R_2(1), R_3(1)\}$

$A = \{R_1 \rightarrow P_1, R_2 \rightarrow P_2, P_2 \rightarrow R_1, P_3 \rightarrow R_2, R_3 \rightarrow P_4, P_1 \rightarrow R_2\}$

Representación matricial

- Se utiliza una matriz de solicitud S , una matriz de asignación A y un vector E de recursos existentes en el sistema.
- Siendo p el número de procesos y r el número de recursos, entonces:
 - A es de tamaño $p \times r$, donde $A[i, j]$ especifica cuantas unidades del recurso j están asignadas al proceso i
 - S es de tamaño $p \times r$, donde $S[i, j]$ especifica cuantas unidades del recurso j está esperando el proceso i
 - E es de tamaño r , donde $E[i]$ especifica cuantas unidades del recurso i existen

Representación matricial

- Se utiliza una matriz de solicitud S , una matriz de asignación A y un vector E de recursos existentes en el sistema.
- Siendo p el número de procesos y r el número de recursos, entonces:
 - A es de tamaño $p \times r$, donde $A[i, j]$ especifica cuantas unidades del recurso j están asignadas al proceso i
 - S es de tamaño $p \times r$, donde $S[i, j]$ especifica cuantas unidades del recurso j está esperando el proceso i
 - E es de tamaño r , donde $E[j]$ especifica cuantas unidades del recurso j existen

Representación matricial

- Se utiliza una matriz de solicitud S , una matriz de asignación A y un vector E de recursos existentes en el sistema.
- Siendo p el número de procesos y r el número de recursos, entonces:
 - A es de tamaño $p \times r$, donde $A[i, j]$ especifica cuantas unidades del recurso j están asignadas al proceso i
 - S es de tamaño $p \times r$, donde $S[i, j]$ especifica cuantas unidades del recurso j está esperando el proceso i
 - E es de tamaño r , donde $E[i]$ especifica cuantas unidades del recurso i existen

Representación matricial

- Se utiliza una matriz de solicitud S , una matriz de asignación A y un vector E de recursos existentes en el sistema.
- Siendo p el número de procesos y r el número de recursos, entonces:
 - A es de tamaño $p \times r$, donde $A[i, j]$ especifica cuantas unidades del recurso j están asignadas al proceso i
 - S es de tamaño $p \times r$, donde $S[i, j]$ especifica cuantas unidades del recurso j está esperando el proceso i
 - E es de tamaño r , donde $E[i]$ especifica cuantas unidades del recurso i existen

Ejemplo de representación matricial y múltiples unidades por recurso

- Ejecución de 3 procesos con 3 recursos R_1 (2), R_2 (3) y R_3 (2)

1. P_1 : solicita($R_1[2]$) → solicita 2 unidades

2. P_2 : solicita($R_2[1]$)

3. P_2 : solicita($R_1[1]$) → se bloquea

4. P_3 : solicita($R_2[1]$)

5. P_3 : solicita($R_2[1]$)

6. P_1 : solicita($R_2[1]$, $R_3[2]$) → se bloquea

- Matriz resultante:

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix} \quad S = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad E = [2 \ 3 \ 2] \quad D = [0 \ 0 \ 2]$$

Ejemplo de representación matricial y una unidad por recurso

- Cuatro procesos (P_1, P_2, P_3 y P_4) y tres recursos ($R_1(1), R_2(1)$ y $R_3(1)$)

1. P_1 : solicita(R_1)
2. P_2 : solicita(R_2)
3. P_2 : solicita(R_1) → se bloquea
4. P_3 : solicita(R_2) → se bloquea
5. P_4 : solicita(R_3)
6. P_1 : solicita(R_2)

- Matriz resultante:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad E = [1 \ 1 \ 1] \quad D = [0 \ 0 \ 0]$$