

Estrategias para detección y recuperación de interbloqueos

Adaptación (ver referencias al final)

Vector de recursos existentes

$E =$

R_1	R_2	R_3	R_4
4	2	3	1

Matriz de asignaciones actuales

$C =$

R_1	R_2	R_3	R_4
0	0	1	0
2	0	0	1
0	1	2	0

Vector de recursos disponibles

$A =$

R_1	R_2	R_3	R_4
2	1	0	0

Matriz de peticiones

$R =$

R_1	R_2	R_3	R_4
2	0	1	0
1	0	1	0
2	1	0	0

Algoritmo para detectar interbloqueos

- Cada recurso está asignado o está disponible

$$\sum_{i=1}^n C_{ij} + A_j = E_j$$

- Sean A y B dos vectores.
 - La relación $A \leq B$ indica que **cada** elemento de A es menor o igual al elemento correspondiente de B .
 - $A \leq B$ si y solo si $A_i \leq B_i$ para $1 \leq i \leq m$
 - m es total de recursos de tipo i .

Algoritmo para detectar interbloqueos

- C_{ij} es el número de instancias del recurso j que están actualmente asignadas al proceso P_i .
- C_{ij} número de instancias del recurso j que desea el proceso P_i .

Algoritmo para detectar interbloqueos

1. Buscar un proceso P_i tal que
 $R \leq A$
 - R la i -ésima fila de R (vector fila)
- En este ejemplo:
 - La fila $R_3 = [2, 1, 0, 0]$ cumple la condición
 - Proceso encontrado: P_3

Vector de recursos disponibles

R_1	R_2	R_3	R_4
2	1	0	0

$A =$

Matriz de peticiones

R_1	R_2	R_3	R_4
2	0	1	0
1	0	1	0
2	1	0	0

$R =$

Algoritmo para detectar interbloqueos

2. Si se encuentra el proceso, sumar la i -ésima fila de C a A

- En este punto P_3 se ejecuta y devuelve todos los recursos.
- Se marca P_3 como proceso completado.
- En este ejemplo:
 - La fila $C_3 = [0, 1, 2, 0]$
 - Se suma $C_3 + A$
 - Nuevo valor de $A = [2, 2, 2, 0]$

Vector de recursos disponibles

R_1	R_2	R_3	R_4
2	1	0	0

$A =$

Matriz de asignaciones actuales

R_1	R_2	R_3	R_4
0	0	1	0
2	0	0	1
0	1	2	0

$C =$

Algoritmo para detectar interbloqueos

1. Proceso se repite en busca de otro procesos P_i tal que $R \leq A$

- En este ejemplo:

- La fila $R_2 = [1, 0, 1, 0]$ cumple la condición. La primera también pero elegí esa.
- Proceso encontrado: P_2

Vector de recursos disponibles

R_1	R_2	R_3	R_4
2	2	2	0

$A =$

Matriz de peticiones

R_1	R_2	R_3	R_4
2	0	1	0
1	0	1	0
2	1	0	0

$R =$

Algoritmo para detectar interbloqueos

2. Si se encuentra el proceso, sumar la i -ésima fila de C a A

- En este punto P_2 se ejecuta y devuelve todos los recursos.
- Se marca P_2 como proceso completado.
- En este ejemplo:
 - La fila $C_2 = [2, 0, 0, 1]$
 - Se suma $C_2 + A$
 - Nuevo valor de $A = [4, 2, 2, 1]$

Vector de recursos disponibles

R_1	R_2	R_3	R_4
2	2	2	0

$A =$

Matriz de asignaciones actuales

R_1	R_2	R_3	R_4
0	0	1	0
2	0	0	1
0	1	2	0

$C =$

Algoritmo para detectar interbloqueos

1. Proceso se repite en busca de otro procesos P_i tal que $R \leq A$

- En este ejemplo:

- La fila $R_1 = [2, 0, 1, 0]$ cumple la condición.
- Proceso encontrado: P_1

Vector de recursos disponibles

R_1	R_2	R_3	R_4
4	2	2	1

$A =$

Matriz de peticiones

R_1	R_2	R_3	R_4
2	0	1	0
1	0	1	0
2	1	0	0

$R =$

Algoritmo para detectar interbloqueos

2. Si se encuentra el proceso, sumar la i -ésima fila de C a A

- En este punto P_1 se ejecuta y devuelve todos los recursos.
- Se marca P_1 como proceso completado.
- En este ejemplo:
 - La fila $C_1 = [0, 0, 1, 0]$
 - Se suma $C_1 + A$
 - Nuevo valor de $A = [4, 2, 3, 1]$

Vector de recursos disponibles

R_1	R_2	R_3	R_4
4	2	2	1

$A =$

Matriz de asignaciones actuales

R_1	R_2	R_3	R_4
0	0	1	0
2	0	0	1
0	1	2	0

$C =$

Algoritmo para detectar interbloqueos

- Todos los procesos se ejecutaron y el sistema no entró en interbloqueo.
 - Todos los procesos se marcaron como completados.
- Si al final de ejecutar el algoritmo, quedan procesos sin marcar, dichos procesos **están interbloqueos**.

Algoritmo para detectar interbloqueos

- Al principio todos los procesos están sin marcar
 1. Buscar un proceso desmarcado P_i para el que la i -ésima fila de R sea menor o igual A .
 2. Si P_i se encuentra, sumar la i -ésima fila de C a A , marcar el proceso y regresar al paso No. 1.
 3. Si P_i no se encuentra, el algoritmo termina y los proceso no marcados están en interbloqueo.

Recuperación de interbloqueo

- **Expropiación**

- Quitar temporalmente un recurso asignado a un proceso y entregarlo a otro proceso.
- No es un procedimiento simple.
- Depende del tipo de recurso.
- Se debe elegir correctamente cuál proceso se suspende para quitar con facilidad un recurso

Recuperación de interbloqueo

- **Retroceso**

- Establecer (guardar) puntos de comprobación de manera periódica.
- Guardar estados del sistema para poder devolverse en caso de un interbloqueo.
- Tomar una instantánea del sistema en un momento dado.
- Guardar la imagen de memoria
- Guardar el estado del recurso: qué recursos están asignados al proceso.
- Se mantiene las instantáneas (no sobrescribir)
- Detección de interbloqueo devolver a un estado anterior y reasignar recursos para evitar la situación de interbloqueo.

Recuperación de interbloqueo

- **Eliminar procesos**

- La estrategia más cruda y simple
- Romper el ciclo: eliminar procesos que estén en un ciclo de interbloqueo
- Elegir un proceso que se pueda eliminar sin efectos dañinos para el sistema
- P. Ej.: proceso que agrega un 1 a un registro en una BD (NO)

Referencias

- Tanenbaum, A. S. (2009). Detección y recuperación de un interbloqueo. In *Sistemas Operativos Modernos* (3rd ed., pp. 442–448). Pearson Educación.