Algoritmo de Maekawa – Ejemplo

Sea el conjunto de procesos p1, p2, p3 y p4 que colaboran en una tarea distribuida y comparten un recurso (CS).

Realice el análisis de votación para el uso de recurso compartido con el algoritmo de Maekawa, para un estado inicial en que todos los procesos tienen el estado RELEASE y su registro de votación es FALSO, dados los siguientes eventos:

Evento 1.- p3 solicita acceso al recurso compartido.

Evento 2.- Algún tiempo después del evento 1, p1 solicita acceso al recurso compartido, en este instante, p3 aún no libera el recurso compartido.

Evento 3.- p3 libera el recurso compartido.

Evento 4.- p1 libera el recurso compartido.

Primer paso – crear la matriz cuadrada de :

N = 4, = 2

Segundo paso – obtener los conjuntos Vi:

Para el proceso p1, el conjunto V1 es:

V1 = {p1, p2, p3}

Para el proceso p2, el conjunto V2 es:

V2 = {p1, p2, p4}

Para el proceso p3, el conjunto V3 es:

V3 = {p1, p3, p4}

Para el proceso p4, el conjunto V4 es:

V4 = {p2, p3, p4}

Tercer paso – realizar el análisis de los eventos:

Considere la siguiente representación de los estados que puede tener un proceso:

R .- RELEASE

W.- WANTED

H.- HELD

También considere la siguiente representación para el estado de votación:

T.- TRUE

F.- FALSE

Evento 0: Estado inicial. Todos los procesos tienen el estado RELEASE y su estado de votación es FALSE.

Evento 1: p3 solicita acceso al recurso compartido, enviando un mensaje a todos los procesos en el grupo V3.

Evento 1.1: Dado que los demás procesos en V1 están en estado RELEASE, y no tienen registro de una solicitud previa, contestan inmediatamente. Los procesos en V1 cambian su estado de votación a TRUE, además, el proceso p3 entra a CS y cambia su estado a HELD.

Evento 2: p1 solicita acceso al recurso compartido enviando una petición a V1, en este instante, p3 aún no libera el recurso compartido. P1 cambia su estado a WANTED.

Evento 2.1: Dado que existe un evento en CS registrado por el propio p1 y por p3, no responden a la solicitud hecha y encolan la petición sin responder. p2 no tiene registro de otra petición anterior, por tanto, responde de inmediato a la solicitud y cambia su estado de votación a TRUE.

Evento 3: p3 libera el recurso compartido. p3 cambia su estado a RELEASE, y envía un mensaje de liberación de CS a los procesos en V3.

Evento 3.1: Al recibir un mensaje de liberación, p4 cambia su estado de votación a FALSE. p3 tiene un mensaje en su cola de peticiones, por tanto, obtiene el primer elemento de la cola, y envía mensaje a p1 concediendo su voto para la entrada a CS. P1 procesa su propia petición y cambia se estado a HELD. Dado que no existen mas peticiones, las colas de petición de los procesos quedan vacías.

Evento 4: p1 libera CS. P1 envía un mensaje de liberación a los procesos en V1, cambia su estado a RELEASE y su estado de votación a FALSE.

Evento 4.1: p2 y p3 reciben el mensaje de liberación y debido a que sus colas de peticiones están vacías, cambian su estado de votación a FALSE.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Evento | P1 | | | P2 | | | P3 | | | P4 | | |
| Estado | Votación | Cola de peticiones | Estado | Votación | Cola de peticiones | Estado | Votación | Cola de peticiones | Estado | Votación | Cola de peticiones |
| 0 | R | F | NULL | R | F | NULL | R | F | NULL | R | F | NULL |
| 1 | R | F | NULL | R | F | NULL | W | T | NULL | R | F | NULL |
| 1.1 | R | T | NULL | R | F | NULL | H | T | NULL | R | T | NULL |
| 2 | W | T | NULL | R | F | NULL | H | T | NULL | R | T | NULL |
| 2.1 | W | T | P1 | R | T | NULL | H | T | P1 | R | T | NULL |
| 3 | W | T | P1 | R | T | NULL | R | T | P1 | R | T | NULL |
| 3.1 | H | T | NULL | R | T | NULL | R | T | NULL | R | F | NULL |
| 4 | R | F | NULL | R | T | NULL | R | T | NULL | R | F | NULL |
| 4.1 | R | F | NULL | R | F | NULL | R | F | NULL | R | F | NULL |