



Universidad Nacional del Este
Facultad Politécnica



Sistemas Distribuidos

Actividad: Asignación de procesos

Alumnos: Mackey Ryuto Sugawara Mochinaga
Oscar Sebastián Domínguez Aranda

Carrera: Ing. de Sistemas

Semestre: 6°



Índice

Introducción	3
Algoritmos centralizados Vs Distribuidos	4
Un algoritmo Centralizado	4
Un Algoritmo Distribuido	4
Bibliografía	6



Introducción

Hoy en día es fácil reunir sistemas de cómputo, compuestos por un gran número de CPUs, conectados mediante una red de alta velocidad, los cuales reciben el nombre genérico de Sistemas Distribuidos, en contraste con los Sistemas Centralizados que constan de un único CPU, sus periféricos de memoria y algunas terminales.

Un sistema distribuido consta de varios procesadores. Estos se pueden organizar como colección de estaciones de trabajo personales, una pila pública de procesadores o alguna forma híbrida. En todos los casos, se necesita cierto algoritmo para decidir cuál proceso hay que ejecutar y en qué máquina.

En el presente trabajo veremos algunos modelos y algoritmos que, con el paso de los años, han propuesto un gran número para la asignación de procesadores con sus aspectos de diseño e implantación y sus principales decisiones que deben tomar los diseñadores y analizando cada una de ellas en su turno.



Algoritmos centralizados Vs Distribuidos

La recolección de toda la información en un lugar permite tomar una mejor decisión, pero menos robusta y coloca una carga pesada en la maquina central. Son preferibles los algoritmos descentralizados, pero se han propuesto algunos algoritmos centralizados por la carencia de alternativas descentralizadas adecuadas.

Un algoritmo Centralizado

La forma mas directa de lograr la exclusión mutua en un sistema distribuido es simular a la forma en que se lleva a cabo en un sistema monoprocesador.

Se elige un proceso coordinador.

Cuando un proceso desea ingresar a una región crítica:

- Envía un mensaje de solicitud al coordinador:
 - Indicando la región crítica.
 - Solicitando permiso de acceso.
- Si ningún otro proceso esta en ese momento en esa región crítica:
 - El coordinador envía una respuesta otorgando el permiso.
- Cuando llega la respuesta el proceso solicitante entra a la región crítica.
Si un proceso pide permiso para entrar a una región crítica ya asignada a otro proceso:
- El coordinador no otorga el permiso y encola el pedido.
Cuando un proceso sale de la región crítica envia un mensaje al coordinador para liberar su acceso exclusivo:
- El coordinador extrae el primer elemento de la cola de solicitudes diferidas y envia a ese proceso un mensaje otorgando el permiso, con lo cual el proceso queda habilitado para acceder a la región crítica solicitada.

Es un esquema sencillo, justo y con pocos mensajes de control.

La limitante es que el coordinador puede ser un cuello de botella y puede fallar y bloquear a los procesos que esperan una respuesta de habilitación de acceso.

Un Algoritmo Distribuido

El objetivo es no tener un único punto de fallo (El coordinador central).

Un ej. Es el algoritmo de Lamport mejorado por Ricart y Agrawala.

Se requiere un orden total de todos los eventos en el sistema para saber cuál ocurrió primero.

Cuando un proceso desea entrar a una región crítica:

- Construye un mensaje con el nombre de la región crítica, su número de proceso y la hora actual.
- Envía el mensaje a todos los demás procesos y de manera conceptual a el mismo.
- Se supone que cada mensaje tiene un reconocimiento.



Universidad Nacional del Este

Facultad Politécnica



Si el receptor no está en la región crítica y no desea entrar a ella, envía de regreso un mensaje o.k. al emisor.

Si el receptor ya está en la región crítica no responde y encola la solicitud.

Si el receptor desea entrar a la región crítica, pero aun no lo logra, compara:

- La marca de tiempo del mensaje recibido con,
- La marca contenida en el mensaje que envió a cada uno.
- La menor de las marcas gana.
- Si el mensaje recibido es menor el receptor envía un o.k.
- Si su propio mensaje tiene una marca menor el receptor no envía nada y encola el pedido.

Luego de enviar las solicitudes un proceso:

- Espera hasta que alguien más obtiene el permiso.
- Cuando llegan todos los permisos puede entrar a la región crítica.

Cuando un proceso sale de la región crítica:

- Envía un mensaje o.k. a todos los procesos en su cola.
- Elimina a todos los elementos de la cola.

La exclusión mutua queda garantizada sin bloqueo ni inanición.

El número de mensajes necesarios por entrada es $2(n-1)$, siendo n el número total de procesos en el sistema.

No existe un único punto de fallo sino n :

- Si cualquier proceso falla no responderá a las solicitudes.
- La falta de respuesta se interpretará como negación de acceso:
 - Se bloquearán los siguientes intentos de los demás procesos por entrar a todas las regiones críticas.

Se incrementa la probabilidad de fallo en n veces y también el tráfico en la red.

Se puede solucionar el bloqueo si:

- El emisor espera y sigue intentando hasta que regresa una respuesta o,
- El emisor concluye que el destinatario está fuera de servicio.

Otro problema es que:

- Se utilizará una primitiva de comunicación en grupo o,
- Cada proceso debe mantener la lista de miembros del grupo, incluyendo los procesos que ingresan, los que salen y los que fallan.
- Se complica para gran número de procesos.

Un importante problema adicional es que:

- Todos los procesos participan en todas las decisiones referentes a las entradas en las regiones críticas.
- Se sobrecarga el sistema

Una mejora consiste en permitir que un proceso entre a una región crítica con el permiso de una mayoría simple de los demás procesos (en vez de todos):

- Luego de que un proceso otorgó el permiso a otro para entrar a una región crítica, no puede otorgar el mismo permiso a otro proceso hasta que el primero libere su permiso.



Bibliografía

- [1 <http://sd-grupo3-asignaciondeprocesadores.blogspot.com/>. [En línea].
]
- [2 http://sistop.gwolf.org/html/biblio/Sistemas_Operativos_-_Luis_La_Red_Martinez.pdf. [En
] línea].
- [3 https://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/2496/6/bastidas_sistemas_operativos_1995.pdf. [En línea].