

Sistemas Operativos

Segundo Cuatrimestre de 2010

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico N°3

Algoritmos en sistemas distribuidos

Grupo

Integrante	LU	Correo electrónico
Mataloni Alejandro	706/07	amataloni@gmail.com
Mancuso Emiliano	597/07	emiliano.mancuso@gmail.com

1. Ejercicio 1

En este trabajo implementaremos un algoritmo para resolver el problema de exclusión mutua en sistemas distribuidos. El prototipo de ejercicio consta de un cliente desarrollado por la cátedra y un servidor el cual tenemos que implementar basándonos en el algoritmo de Ricart-Agrawala.

Para realizar esta tarea, nos ocuparemos de manejar los mensajes enviados cliente-servidor, así como también con los servidor-servidor. Cada cliente tiene asignado un servidor y solo puede enviar y recibir mensajes del mismo. Sin embargo un servidor se comunica con su único cliente pero también con el resto de los servidores.

El servidor básicamente se encuentra recibiendo mensajes de cualquier tipo. Cuenta con una serie de variables útiles a para definir su comportamiento:

- int seqnrorecibida;
- int ourSeqNumber;
- int highestSeqNum;
- int outstandingReplycount;
- bool requestingCriticalSection;
- bool replyDeferred[cantserv];
- bool deferIt;

Hay 2 grandes tipos de mensajes:

- cliente-servidor
- servidor-servidor

1.1. Mensajes cliente-servidor

Mensaje tipo **PEDIDO**:

- El cliente le pide al servidor, acceso a la sección crítica.
- El servidor le responde al cliente cuando puede acceder a su sección crítica. También envía un mensaje de tipo **SERVPED** al resto de los servidores.

Mensaje tipo **LIBERO**:

- El cliente le avisa al servidor, que sale de la sección crítica.
- El servidor asociado, responde a los pedidos de los servidores pendientes enviando un mensaje de tipo **SERVRES**.

Mensaje tipo **TERMINO**:

- El cliente le avisa al servidor, que termina su ejecución.

1.2. Mensajes servidor-servidor

Mensaje tipo **SERVPED**:

- Al enviar, se pide permiso para acceder a la sección crítica.
- Al recibir, se le responde inmediatamente con un mensaje de tipo **SERVRES**, o en caso de tener prioridad sobre el otro servidor, se pospone la respuesta.

Mensaje tipo **SERVRES**:

- Al enviar, el servidor responde al pedido de la sección crítica.
- Al recibir, si se tienen las respuestas de todos los servidores, se le da el permiso al cliente asociado para entrar a la sección crítica a través de un mensaje de tipo PEDIDO.

2. Pruebas realizadas

Más allá de hacer muchos *printf* para ver que nuestro sistema estaba haciendo lo que debería, nosotros quisimos asegurarnos de que funcionaba correctamente. Hicimos lo siguiente, creamos una prueba propia en donde 2 clientes se disputarían la sección crítica. Un cliente empieza pidiendo acceder a la sección crítica y no liberarla por un largo rato, en el medio el otro cliente hace el pedido. Los archivos que escribimos fueron los siguientes:

Cliente 1	Cliente 2
pedir	nada
nada	pedir
nada	escribir Yo tengo el poder!
nada	liberar
nada	
nada	
nada	
liberar	

Aparte de esto también modificamos un poco el código del cliente. Lo que hicimos fue que cuando se lee una instrucción de *nada* el cliente se mande a dormir un tiempo. Esto es para que el cliente que primero accede a la sección crítica luego se quede haciendo *sleep* en las siguientes operaciones así el otro cliente pide la sección crítica y se queda esperando hasta que el primero no termine de hacer todos los *nada* que tiene.