Introducción al Modelado de Sistemas
Proyecto Final

Profesores: Carlos Ramírez y Camilo Rocha

2 de mayo de 2017

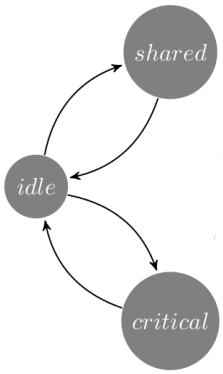
En esta ocasión se modelará un protocolo de coherencia de caché llamado ESI (del inglés, *exclusive*, *shared*, *inactive*). Este protocolo es importante porque ha sido usado en la implementación de políticas de acceso a la memoria en un computador. El interés en este protocolo, desde el punto de vista de los métodos formales, radica en que permite mantener la integridad de bloques de memoria cuando varios procesos tienen acceso concurrente para leer o escribir en ellos. ESI debe servir para una cantidad arbitraria de procesos y de direcciones de memoria, lo cual presenta un reto para las técnicas de verificación tradicionales.

La especificación de ESI en este proyecto está dividida en dos partes. En la primera parte se modelarán aspectos estructurales del protocolo con detalles de la dinámica de acceso a las direcciones de memoria. En la segunda parte se agregarán detalles de cómo pueden cambiar los valores almacenados en los procesos (i.e., el caché) y en la memoria principal.

Parte 1

En ESI hay una cantidad arbitraria de procesos y de direcciones de memoria. Un proceso puede estar en uno de tres estados: inactivo, compartido, exclusivo. Un proceso en estado *inactivo* no tiene acceso a ninguna dirección de memoria y, en consecuencia, no puede ni leer ni escribir desde/en ninguna dirección de memoria. Un proceso en estado *compartido* tiene permiso de lectura para una dirección de memoria dada. Un proceso en estado *exclusivo* tiene permiso de lectura y escritura para una dirección de memoria dada. Un proceso puede tener acceso a más de una dirección de memoria en el sistema y en cada caso, posiblemente, con permisos distintos. Por ejemplo, el proceso p_0 puede tener acceso a las memorias m_1, m_8, m_9: para m_1 y m_8 con acceso de lectura y para m_9 con acceso de (lectura y) escritura. Además, una memoria puede tener varios procesos asociados a ella.

El siguiente autómata muestra cómo puede un proceso cambiar de estado para una dirección de memoria dada. Las palabras 'idle', 'shared' y 'critical' en esta gráfica representan, respectivamente, los estados inactivo, compartido y exclusivo. Por ejemplo, para una dirección de memoria dada, un proceso puede pasar de inactivo a compartido o de compartido a inactivo. Sin embargo, un proceso nunca puede pasar de compartido a exclusivo.



Para que un proceso pueda tener persmiso de lectura para una memoria m, es necesario que no haya proceso alguno con permiso de escritura para m (pero m puede tener otros procesos asociados con permiso de lectura). A su vez, para que un proceso pueda obtener permiso de escritura para una memoria m, es necesario que m no tenga asociado proceso alguno.

En este sistema es importante mantener las siguientes propiedades, para cada memoria:

1. Hay a lo sumo un proceso con acceso exclusivo.
2. Cuando haya al menos un proceso con acceso compartido, no hay procesos con acceso exclusivo.

¿Qué se observa? En este modelo se observan el comportamiento de los procesos con respecto a sus permisos de acceso a diferentes direcciones de memoria y la dinámica de sus transiciones: (1) qué tipo de permiso tiene un proceso para cada memoria; (2) qué procesos tienen permiso de lectura para cada memoria; (3) qué procesos tienen permiso de escritura para cada memoria.

Parte 2

En la segunda parte del proyecto se aumenta el detalle con el cual se observa el modelo propuesto anteriormente. En particular, se quiere observar qué valor tiene almacenado localmente cada proceso (i.e., caché) y qué valor tiene almacenada cada dirección de memoria. Además, se quiere conocer: (1) cuántas veces ha sido leída una memoria; (2) cuántas veces ha sido modificada una memoria; (3) cuál ha sido la mayor cantidad de procesos que ha accedido simultáneamente con permisos de lectura una memoria; (4) cuál ha sido la mayor cantidad de procesos que ha accedido simultáneamente con permisos de escritura una memoria (note que este valor debe ser 0 o 1). Cuando un proceso tiene permiso de lectura para una memoria m, este puede copiar el valor almacenado en m a su copia local. Cuando un proceso tiene permiso de escritura para una memoria m, este puede modificar el valor almacenado en m con el valor de su copia local. En esta parte, no es posible que un proceso obtenga permiso de lectura o escritura para una dirección de memoria cuando el valor almacenado en dicha memoria es igual al valor de su copia local. Tenga en cuenta que la copia local de un proceso puede cambiar arbitrariamente cuando esté inactivo.