

### **Detalles adicionales sobre los LW's y SW's**

En el archivo Documentación.pdf que se encuentra en la misma carpeta que este archivo se describe con detalle mediante pseudocódigo la lógica a seguir para ejecutar las instrucciones load y store.

Sin embargo no se hace énfasis en los bloqueos y como los utilizamos, por lo que en este documento se mostrará de forma general como hacemos la sincronización.

#### **LOAD**

- Intenta bloquear la caché local hasta que lo logre, si el bloque solicitado está en caché modificado o compartido, hace el load, libera la caché y termina la instrucción.
- Si el bloque que se necesita no está en su caché en los estados mencionados anteriormente se sigue así:
  - .1. Se libera la caché local.
  - .2. Intenta bloquear la el directorio correspondiente hasta que lo logre.
  - .3. Intenta bloquear la caché local para traerlo de memoria, si no lo logra libera el directorio y vuelve al punto 2. Si lo logra trae de memoria, hace el load, libera la caché local y el directorio y termina.

#### **STORE**

- Intenta bloquear la caché local hasta que lo logre, si el bloque solicitado está en caché modificado, hace el store, libera la caché y termina la instrucción.
- Si el bloque que se necesita no está en su caché y está modificado por otro bloque.
  - .1. Se libera la caché local.
  - .2. Intenta bloquear la el directorio correspondiente hasta que lo logre.
  - .3. Intenta bloquear la caché que tiene el bloque modificado para escribirlo a memoria, si no lo logra libera el directorio y vuelve al punto 2. Si lo logra escribe a memoria, lo pone compartido y libera la caché que tenía el bloque modificado.

Ahora intenta bloquear la caché local para traerlo de memoria, si no lo logra libera el directorio y vuelve al punto 2. Si lo logra trae de memoria, hace el store, libera la caché local y el directorio y termina.
- Si el bloque que se necesita no está en su caché y está compartido por otros bloques.
  - .1. Se libera la caché local.
  - .2. Intenta bloquear la el directorio correspondiente hasta que lo logre.
  - .3. Intenta bloquear la primer caché que tiene el bloque compartido para invalidarla, si no lo logra libera el directorio y vuelve al punto 2. Si lo logra la invalida, la libera y trata de hacer lo mismo con la siguiente caché compartida.

Una vez que ha invalidado las otras cachés, el único bloqueo que tiene es el del directorio, así intenta bloquear la caché local para traerlo de memoria, solo trae de memoria en caso de que no estuviera compartido por la caché local también; si no lo logra libera el directorio y vuelve al punto 2. Si lo logra trae de memoria, hace el store, libera la caché local y el directorio y termina.

**En todos los casos solo hay 2 recursos bloqueados y siempre se liberan si no se obtienen los 2 bloqueos por lo que evitamos cualquier posible deadlock.**