
Capítulo 8

MEMORIA CACHÉ

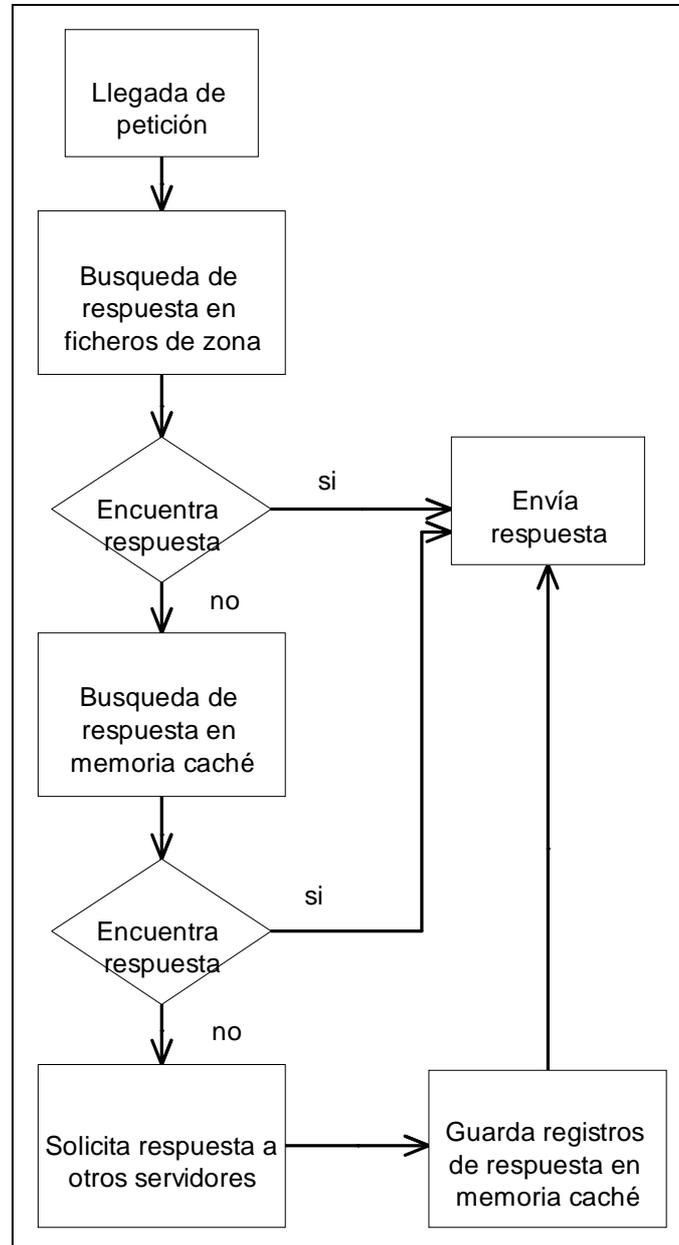
Este capítulo trata sobre la memoria caché que se encuentra implementada dentro del sistema ANS. Se describe la funcionalidad de la memoria caché dentro del sistema ANS, así como la estructura interna que tiene y cómo ha sido implementada.

DESCRIPCIÓN MEMORIA CACHÉ EN SISTEMA ANS

El sistema ANS está formado por una serie de máquinas que se comunican entre sí a través de la red ATM. Esta comunicación entre distintas máquinas se produce al tener que solicitar información acerca de un nombre de dominio del que no se tienen datos en la máquina local. Todo este flujo de información produce carga de datos en la red, así como un retraso en la obtención de la misma, siendo preferible el tener estos datos en la máquina local, sin tener que consultar a otra máquina para obtenerlos. Para intentar disminuir un poco la carga de tráfico que el protocolo ANS cursa por la red y disminuir tiempos de respuesta se crea la memoria caché.

La memoria caché, tal y como su nombre indica, es una zona de memoria de la máquina local en la que se van a ir almacenando *registros fuentes* que se han ido obteniendo a través de peticiones a otras máquinas del sistema ANS. Estos *registros fuentes*, como es lógico, no se encuentran en los ficheros de datos locales que posee la máquina. De esta forma, al intentar resolver una petición, primeramente se buscará en los ficheros de datos locales la respuesta a dicha petición, y en caso de no encontrarla, se buscará en la memoria caché una posible respuesta a dicha petición.

En el siguiente diagrama se muestra el funcionamiento normal de un servidor a la hora de buscar respuesta a una petición:



El tiempo de permanencia de un *registro fuente* dentro de la memoria caché se rige por un campo del mismo denominado TTL(Time To Live). Este es el tiempo máximo que puede permanecer un registro fuente en la memoria caché, pasando a invalidarse su contenido a partir de este tiempo. Otra situación a tener en cuenta es que puede ocurrir que la memoria caché se llene de registros fuentes, y a ninguno de ellos le haya expirado su TTL. En este caso, para introducir un nuevo registro fuente dentro de la memoria caché, se sacará de la misma aquél registro fuente que le quede menos tiempo para que expire su TTL.

IPC

Las IPC(Inter Process Communication) son unas utilidades que ofrece el sistema UNIX para facilitar el intercambio de información entre procesos dentro de una misma máquina. Las tres grandes utilidades que forman las IPC son:

- **MEMORIA COMPARTIDA:** Permite la reserva de memoria, la cual puede ser compartida por más de un proceso dentro de la misma máquina.
- **SEMÁFOROS:** Permite hacer un uso exclusivo de un determinado recurso.
- **MENSAJES:** Permite el intercambio de información en forma de mensajes entre distintos procesos.

El sistema ANS hará uso de las utilidades de memoria compartida y la de semáforos. Desde el punto de vista del sistema ANS la memoria caché será una zona de memoria compartida, a la cual tendrán acceso los procesos que estén resolviendo peticiones. El acceso a dicha zona de memoria debe ser limitado, ya que no es posible que un proceso lea de la memoria caché y, a la vez, otro la esté modificando. Para la solucionar este problema de acceso exclusivo se hace uso de los semáforos, haciendo posible que cuando algún proceso acceda a la memoria caché, ningún otro pueda modificarla.

Las funciones que nos ofrece las utilidades de memoria compartida y semáforo IPC se describen a continuación:

MEMORIA COMPARTIDA

Una zona de memoria compartida se identifica dentro de una misma máquina con un número entero positivo. Con este número entero positivo y las funciones ofrecidas por las IPC, se podrá hacer determinar unívocamente una zona de memoria compartida. Las funciones que ofrecen las IPC para el manejo de memoria compartida son las siguientes:

- *shmget()*: Realiza un enlace de la zona de memoria compartida al proceso. Como parámetro se le pasa el identificador(número entero positivo) de la memoria caché. Si la zona de memoria compartida no está creada, se puede especificar que la cree.
- *shmat()*: Devuelve un puntero que apunta al comienzo de la zona de memoria compartida.
- *shmdt()*: Desliga el puntero devuelto por *shmat()* de la zona de memoria compartida.
- *shmctl()*: Realiza una operación de control sobre la zona de memoria compartida. Con esta función, entre otras cosas, se puede liberar la zona de memoria compartida.

SEMÁFOROS

Los semáforos son una utilidad de las IPC que permiten el uso ó acceso exclusivo de un determinado recurso del sistema. Un semáforo se identifica con un contador sobre el que se pueden realizar las operaciones de incremento, decremento o consulta. En función del valor que tenga el semáforo en el momento de realizar una determinada operación, esta provocará que el proceso que ejecutó la operación se bloquee o continúe con la ejecución.

<i>Valor semáforo</i>	<i>Operación</i>	<i>Acción</i>
>0	Incremento	Se incrementa el semáforo, y continúa ejecución del proceso.
	Decremento	Se decrementa el semáforo, y continúa con la ejecución del proceso.
	Consulta	Se bloquea el proceso, hasta que el semáforo alcance un valor de 0.
<0	Incremento	Se incrementa el semáforo, y continúa ejecución del proceso.
	Decremento	Se bloquea el proceso, hasta que el semáforo alcance un valor mayor a 0.
	Consulta	Se bloquea el proceso, hasta que el semáforo alcance un valor de 0.
=0	Incremento	Se incrementa el semáforo, y continúa ejecución del proceso.
	Decremento	Se bloquea el proceso, hasta que el semáforo alcance un valor mayor a 0.
	Consulta	Se continúa con la ejecución del proceso.

Al igual que la memoria compartida, los semáforos se identifican dentro de la una máquina con un número entero positivo. Las funciones que ofrecen las IPCs para el manejo de semáforos son las siguientes:

- *semget()*: Crea un enlace entre el semáforo y el proceso que llama a esta función. Si el semáforo no existe lo crea.
- *semctl()*: Realiza una operación de control sobre el semáforo(asignarle un valor al semáforo, eliminar el semáforo, etc...)
- *semop()*: Realiza una operación sobre el semáforo(incremento, decremento, consulta).

Una cosa que cabe destacar es que las IPCs permiten el manejo de conjuntos o *arrays* de semáforos.

APLICACIÓN AL SISTEMA ANS

La memoria caché se implementa dentro del sistema ANS como una zona de memoria compartida que estará formada por una serie de celdas, celdas que a su vez contendrán los registros fuentes almacenados dentro de la memoria caché. En el **ANEXO IV** de esta memoria se entra más en detalle de la estructura interna de la memoria caché y de las tareas propias de la misma.

Para el manejo de la memoria caché, el sistema ANS utiliza la clase *gestor_caché* que ofrece como métodos todas las operaciones básicas que se pueden realizar con la memoria caché. Las operaciones que ofrece esta clase se describen a continuación:

- *inicializa_caché_primaria()*: Crea la zona de memoria caché y el semáforo que rige el acceso a la misma. Sólo puede ser ejecutado por un proceso dentro de la misma máquina, por lo que se ejecutará al comienzo de la aplicación servidora.
- *inicializa_caché_secundaria()*: Crea enlace a la zona de memoria caché y al semáforo previamente creados. Lo ejecutan los procesos que quieren tener acceso a la memoria caché y no siendo el proceso principal de la aplicación servidora.
- *mete_registro()*: Inserta un registro fuente dentro de la memoria caché.
- *lee_registro()*: Lee de forma secuencial los registros fuentes que hay en la memoria caché.
- *actualiza_caché()*: Elimina los registros cuyos TTL hayan expirado dentro de la memoria.
- *reorganiza_caché()*: Reordena las celdas que forman la memoria caché, y las coloca secuencialmente en la zona de memoria caché.

Cabe destacar el mecanismo utilizado para la obtención del acceso exclusivo por parte de los distintos procesos dentro de la memoria caché. Las reglas que debe cumplir dicho acceso son las siguientes:

1. Mientras un proceso lea dentro de la memoria caché, ningún otro proceso puede modificar la memoria caché, aunque sí puede leer de la misma.
2. Mientras un proceso modifica la memoria caché, ningún otro proceso puede ni leer ni modificar la memoria caché.

Estas reglas se implementan mediante un array de dos semáforos, uno denominado de *lectura*, y otro denominado de *escritura*. Los accesos a la memoria caché se dividen en dos tipos, de lectura(para obtener datos de la memoria caché), y de escritura(para modificar la memoria caché). Partimos de la base de colocar ambos semáforos a valor cero. Las operaciones a realizar sobre el array de semáforos en cada tipo de acceso son las siguientes:

Acceso Lectura:

- Al entrar:*
1. Se incrementa el semáforo de lectura(se indica que hay alguien leyendo).
 2. Se consulta el semáforo de escritura(se consulta si hay alguien escribiendo).
- Al salir:*
1. Se decrementa el semáforo de lectura(se indica que alguien ha dejado de leer).

Acceso Escritura:

- Al entrar:*
1. Se consulta el semáforo de lectura(se consulta si hay alguien leyendo).
 2. Se consulta el semáforo de escritura(se consulta si hay alguien escribiendo).
 3. Se incrementa el semáforo de escritura(se indica que hay alguien escribiendo).
- Al salir:*
1. Se decrementa el semáforo de escritura(se indica que alguien ha dejado de escribir).

Con las anteriores operaciones sobre los semáforos se consigue que no se creen inconsistencias debido al acceso simultáneo por parte de varios procesos a la zona de memoria caché.