
Capítulo 11

APLICACIÓN SERVIDORA

En este capítulo se describe la aplicación ejecutada en una máquina servidora y que ofrece el servicio ANS. También trata sobre el funcionamiento de la máquina de estados servidora, utilizada por dicha aplicación, encargada de la resolución de las peticiones que van llegando al servidor.

INTRODUCCIÓN

Un servidor dentro del sistema ANS se compone fundamentalmente de un proceso principal que es el encargado de tareas como son la creación y mantenimiento de la memoria caché, la actualización de las distintas zonas del espacio de nombres que lleve el servidor y la recepción de peticiones por parte de clientes. Una particularidad es que el proceso servidor, cada vez que recibe una petición, lanza un proceso hijo que será el encargado de su resolución. Con este mecanismo se consigue el aumento de la capacidad de una máquina servidora, ya que así puede resolver varias peticiones en paralelo.

Para la resolución de las peticiones que le llegan al servidor, el sistema ANS implementa una máquina de estados. Esta máquina de estados tiene definida una serie de estados por los que atravesará una petición durante el proceso de resolución de la misma. Estos estados dependerán de la petición en sí misma y de los resultados que vayan produciendo las distintas búsquedas realizadas para la resolución de dicha petición.

En el sistema ANS, la máquina de estados servidora de nombres se implementa como una llamada a una determinada función, de forma que para el proceso que ejecute dicha máquina de estados lo que ocurra en su interior le es indiferente, sólo importándole el resultado que de ella se obtiene.

DESCRIPCIÓN DE LA MÁQUINA DE ESTADOS

Las peticiones que le llegan a un servidor del sistema ANS son resueltas por éste mediante una máquina de estados. Esta máquina de estados servidora de nombres implementa las tareas propias de la resolución de una petición. Estas tareas se descomponen en:

- Búsqueda de la respuesta a la petición en los datos locales del servidor de nombres (ficheros de datos de zonas, memoria caché).
- Solicitud de respuesta a la petición a otros servidores de nombres del sistema ANS.
- Análisis de las respuestas obtenidas de otros servidores de nombres.
- Devolución de un mensaje de respuesta con la información solicitada.

La máquina de estados servidora se describe de una forma detallada en el **ANEXO VI** de esta memoria. En el sistema ANS la máquina de estados servidora está implementada como una función, a la cual se le pasa como parámetro una determinada petición, y se obtiene un mensaje de respuesta, haciendo todas las tareas de búsqueda de información y de comunicación con otros servidores transparentes para la aplicación que ejecutó la máquina de estados.

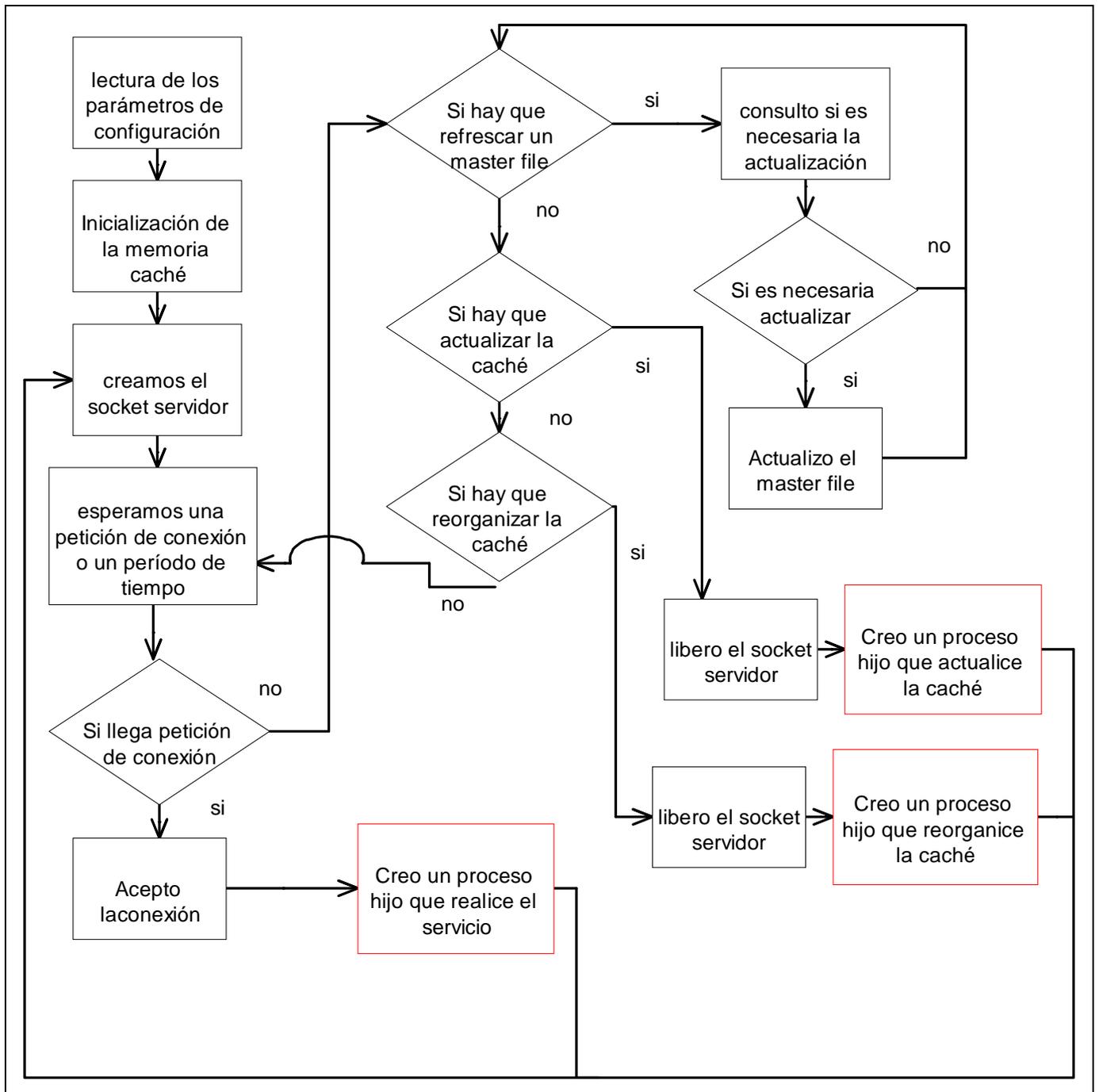
Un aspecto a reseñar sobre la máquina de estados servidora de nombres, es que para el caso de resolución de peticiones con el método recursivo se hace uso de la máquina de estados *RESOLVER*. Se recuerda que un *RESOLVER* era una máquina de estados que se encargaba de buscar información en local acerca de una petición determinada, y en caso de no encontrarla realizaba consultas a otros servidores para recabar la respuesta a dicha petición. Esto lo que indica es que un servidor a la hora de resolver una petición también funciona como cliente de algún otro servidor.

DESCRIPCIÓN APLICACIÓN SERVIDORA

El servidor de nombres del sistema ANS se compone principalmente de un proceso o aplicación principal que es la encargada de un conjunto de tareas básicas:

- Creación y mantenimiento de la memoria caché. En la aplicación principal se crea la zona de memoria compartida denominada memoria caché, y cada cierto tiempo (configurable como parámetro) se van a realizar tareas de actualización y de reorganización de la caché.
- Tareas de mantenimiento de los ficheros de zonas de datos. La aplicación principal es la encargada de implementar el protocolo de actualización de ficheros de datos de zona. Será la encargada de gestionar los tiempos de refresco, reintento y expiración que contempla el protocolo de actualización.
- Creación y gestión del socket servidor. Esta aplicación maneja el socket servidor, por el cual la aplicación servidora está siempre a la espera de peticiones. Cada vez que le llegue una determinada petición, la aplicación principal lanzará un nuevo proceso hijo que será el que gestione la petición.

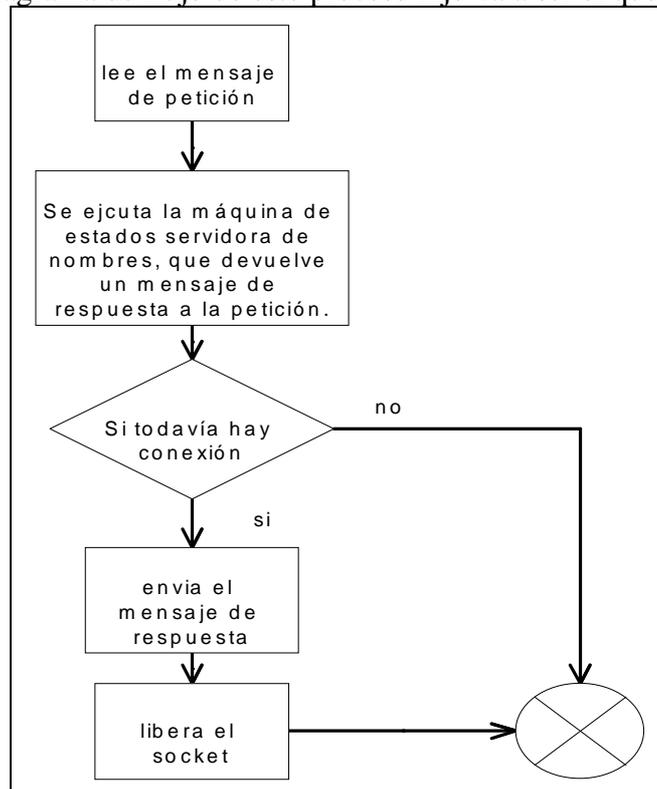
El diagrama de flujo que sigue este ejecutable es un poco complejo:



Se explica el funcionamiento de diagrama anterior:

1. Primeramente se realiza la lectura de los parámetros de configuración del sistema, así como la inicialización de un objeto CACHÉ, que iniciará la memoria caché.
2. Se crea un socket servidor, por el que se permanece a la espera de una petición de establecimiento de conexión.
3. Permanece a la escucha por ese socket servidor, hasta máximo de un período de tiempo(parámetro de configuración), si es que no llega anteriormente ninguna petición.
 - 3.1. Si no llega ninguna petición de conexión
 - 3.1.1. Pasa a ver si es necesario un refresco de algún master file, y si es así se procede a su actualización.
 - 3.1.2. Pasa a ver si es necesaria la actualización de la memoria caché, en cuyo caso se cierra el socket servidor, crea un proceso hijo, y en este proceso hijo, actualiza la memoria caché. Vuelve al punto 2.).
 - 3.1.3. Pasa a ver si es necesaria la reorganización de la memoria caché, en cuyo caso se cierra el socket servidor, crea un proceso hijo, y en este proceso hijo, reorganiza la memoria caché. Vuelve al punto 2.).
 - 3.1.4. Si no ha llegado a crear ningún proceso hijo, vuelve al punto 3.).
 - 3.2. Si llega una petición de conexión
 - 3.2.1. Pasa a aceptar la petición de conexión, y crea un proceso hijo que trate esta conexión. El proceso hijo será el que ejecute la máquina de estados servidora de nombres.

El diagrama de flujo de este proceso hijo va a ser el que sigue:



Las tareas que realiza el proceso hijo que implementa el servicio son: lectura del mensaje de petición, ejecución de la máquina de estados servidora de nombres que devuelve un mensaje de respuesta y devolución del mensaje de respuesta a través de la conexión previamente establecida.