

Sea la llamada `RMDIR`, que dado un nombre de directorio comprueba que está vacío y, en caso afirmativo, borra dicho directorio. Esta llamada implica la manipulación de muchas de las estructuras de datos (metadatos) del sistema de ficheros.

Considerando un sistema de ficheros tipo UNIX tradicional (basado en nodos-i) y que los metadatos se escriben siempre utilizando una política de escritura *write-through*, conteste razonadamente a las siguientes cuestiones:

1. Dibuje un esquema de las citadas estructuras de datos del sistema de ficheros involucradas en la llamada `rmdir("/directorio")`; donde muestre el contenido más relevante de cada estructura de datos para la ejecución de dicha llamada (por ejemplo, el valor del campo “Número de enlaces” del nodo-i del directorio a eliminar).
2. En relación a las estructuras del diagrama anteriores, enumere los pasos que se deben realizar en la operación `rmdir("/directorio")`. Para cada paso, indique qué estructura de datos se lee, de dónde se lee (disco, memoria), cuál se modifica en memoria y/o se escribe a disco. Para ello, rellene una tabla como la siguiente (el primer paso es sólo un ejemplo; no tiene por qué corresponder con la operación que se debe analizar):

Paso	Estructura del S.F	Descripción de la acción realizada	Tipo de manipulación de la estructura (Lectura, Modificación en memoria y/o escritura)	Atributo(s) de la estructura modificados
1	Bitmap de bloques	Poner a ocupado el bloque X	Lectura del bitmap de disco, modificación en memoria y escritura en disco	Valor del bit X

3. Durante la ejecución de la llamada `rmdir("/directorio")`, considérese la posible caída del sistema y su efecto sobre la consistencia del sistema de ficheros. Para ello, de acuerdo a los pasos indicados, indique qué problemas de inconsistencia se producirían si el sistema cae justo antes de cada paso.
4. ¿El sistema de ficheros JFS de IBM tiene los problemas de inconsistencia vistos en el apartado anterior? ¿Por qué? Responder razonadamente.

Solución

2.1

Dada la llamada `rmdir("/directorio")`, lo primero que habrá que hacer es una decodificación del directorio para acceder al nodo-i del mismo. En este caso, el nombre es un nombre absoluto, y por tanto, a partir del directorio raíz. Por consiguiente, estará involucrado el nodo-i del directorio raíz (nodo-i 2), que entre sus entradas (en alguno de sus bloques de datos), tendrá la del directorio en cuestión. En caso contrario, la operación de borrado de directorio producirá un error. También se produce un error en el caso de que el usuario que esté invocando el servicio, no tenga los permisos correspondientes para acceder o borrar el directorio.

La estructura del nodo-i 2 será la siguiente:

Nodo-i 2

Tipo fichero = directorio
Nº enlaces = N
Id propietario = root

Gid propietario = root
Permisos
Tamaño
Fecha de creación
Fecha de última modificación
...
Punteros a bloques de datos

Habitualmente el propietario del directorio raíz es el superusuario o *root*. Los punteros a bloques de datos direccionarán los bloques de datos del directorio raíz, que contendrá la siguiente estructura:

“.”	2
“..”	2
.....
“directorio”	15
.....

Se ha considerado que el nodo-i correspondiente a “/directorio” es el 15. El nodo-i 15 tendrá la siguiente estructura:

Nodo-i 15

Tipo fichero = directorio
Nº enlaces = 2
Id propietario
Gid propietario
Permisos
Tamaño
Fecha de creación
Fecha de última modificación
....
Punteros a bloques de datos

Los punteros a bloques de datos direccionarán los bloques de datos del directorio “/directorio”. Supongamos que el bloque de datos correspondiente al directorio es el 2015. Este bloque debe contener la siguiente estructura para estar vacío:

“.”	15
“..”	2

Es decir, las entradas “.” y “..”. Como se puede comprobar, en este caso, el número de enlaces del directorio será igual a 2, correspondientes a las dos entradas que hacen referencia al nodo-i 15.

A la hora de borrar el directorio, primero se deberá comprobar que está vacío, tal y como se ha indicado previamente.

Además, el directorio raíz también quedaría modificado de la siguiente forma:

a) Se elimina la entrada correspondiente al directorio a borrar. Esto podría provocar un cambio en el número de bloques de datos utilizados por dicho directorio, pudiéndose reducir en 1 en caso necesario. No obstante, este

- hecho es improbable. En caso afirmativo, se modificaría la zona de los punteros a los bloques de datos.
- b) Se modificaría el número de enlaces, decrementándolo en 1.
 - c) Se modificaría el tamaño.
 - d) Se modificaría la fecha de última modificación.

El borrado de un directorio también provocaría la modificación de determinadas estructuras del sistema de ficheros, como son los mapas de bits (de bloques y de nodos-i). La estructura del sistema de ficheros es la siguiente:

Boot	Superbloque	Mapas de bits	Nodos-i	Datos y directorios
------	-------------	---------------	---------	---------------------

Los mapas de bits y bloques se modificarían poniendo a 0 (libre) los bits correspondientes al nodo-i en cuestión (en el ejemplo, el 15) y al bloque de datos (en el ejemplo, el 2015).

No se van a considerar en el desarrollo del problema las estructuras correspondientes a los descriptores de los directorios en el BCP del proceso que realiza la operación, ni las tablas intermedias. Por supuesto, al iniciar la operación sería necesario abrir los descriptores y las entradas correspondientes y al finalizar cerrar dichos descriptores y entradas.

2.2

Paso	Estructura del S.F.	Descripción de la acción realizada	Tipo de manipulación de la estructura	Atributo(s) de la estructura modificado(s)
1	Nodo-i 2	Lectura del nodo-i 2 y obtención del bloque de datos asociado	Se supone que el nodo-i 2 está en memoria. Se lee de memoria.	Ninguno
2	Bloque de datos del directorio raíz	Se lee el bloque de datos del directorio raíz, hasta encontrar la entrada “directorio”. En caso de no encontrarse, error y finalizar. En otro caso, se obtiene el nodo-i del directorio (15)	Lectura de disco del bloque de datos, si no está en la cache	Ninguno
3	Nodo-i 15	Se lee el nodo-i 15 y se obtiene el bloque de datos asociado (2015). Previamente, se comprueba que el usuario tiene permisos de lectura y escritura, ya que va a borrar el directorio. Si no tiene permisos, error y finalizar.	Lectura del disco del nodo-i 15, si no está en la cache.	Ninguno

		También se comprueba que es un directorio.		
4	Bloque de datos 2015	Se lee el bloque de datos 2015 y se comprueba que está vacío. Para ello, se comprueba que sólo tiene las entradas “.” y “..”	Lectura del disco del bloque de datos 2015.	Ninguno
5	Bitmap de bloques	Se pone a 0 (libre) el bit correspondiente al bloque 2015.	Lectura del bitmap de bloques de disco, escritura en memoria y en disco (<i>write-through</i>)	Bit 2015 del bitmap de bloques
6	Bitmap de nodos-i	Se pone a 0 (libre) el bit correspondiente al nodo-i 15.	Lectura del bitmap de nodos-i de disco, escritura en memoria y en disco (<i>write-through</i>)	Bit 15 del bitmap de nodos-i.
7	Bloque de datos del directorio raíz	Se elimina la entrada <directorío, 15>. En caso necesario, se modifica algún puntero a bloque de datos, liberándose el bloque de datos correspondiente. Eso supondría también modificar el bitmap de bloques. Esto es improbable.	El bloque de datos probablemente esté en la cache. En dicho caso, modificación en memoria y si se utiliza escritura retardada, no se escribe inmediatamente en disco. En otro caso, lectura de disco y modificación en memoria.	Bloque de datos del directorio raíz. En caso de eliminación de un bloque de datos, modificación del bitmap de bloques.
8	Nodo-i 2	Modificación del número de enlaces. Se decrementa en 1.	Lectura y modificación en memoria, escritura en disco (<i>write-through</i>)	Número de enlaces del nodo-i 2.

9	Nodo-i 2	Modificación del tamaño en caso necesario. Se decrementa con el tamaño de la entrada del directorio borrado.	Lectura y escritura en memoria y en disco (<i>write-through</i>)	Campo tamaño del nodo-i 2.
10	Nodo-i 2	Modificación de la fecha de última modificación con la fecha actual.	Lectura y escritura en memoria y en disco (<i>write-through</i>)	Fecha de última modificación del nodo-i 2.

2.3

Antes del paso 5, no se produciría ninguna inconsistencia, ya que no se ha modificado ninguna estructura. Si el sistema cayera justo antes del paso 6, el bit 2015 del bitmap de bloques indicaría que dicho bloque está libre, y sin embargo, está referenciado por el nodo-i 15. Si ese bloque se le asignara a algún fichero, dos ficheros estarían haciendo referencia al mismo bloque. Éste es un problema muy grave.

Si el sistema cayera justo antes del paso 7, el bit 15 del bitmap de nodos-i indicaría que dicho nodo-i está libre, y sin embargo, está referenciado por la entrada correspondiente en el directorio raíz.

Si el sistema cayera justo antes del paso 8, el problema reside en que no se han modificado correctamente las propiedades del nodo-i 2, es decir, el número de enlaces, el tamaño y la fecha de última modificación.

Si el sistema cayera justo antes del paso 9, no se han modificado correctamente el tamaño y la fecha de última modificación del nodo-i 2.

Si el sistema cayera justo antes del paso 10, no se ha modificado la fecha de última modificación del nodo-i 2.

Si el sistema cayera al finalizar el paso 10, la operación rmdir se ha realizado correctamente.

2.4

El sistema de ficheros JFS de IBM no tiene los problemas de inconsistencia anteriores, puesto que utiliza un modelo de control transaccional basado en *logging*. De este modo, toda suboperación que modifique los metadatos y datos de un archivo se agrupa en la misma transacción. Si el sistema falla, las acciones parcialmente realizadas se deshacen o completan, recorriendo el *log* donde se han anotado las operaciones. No se garantiza que el sistema esté actualizado al finalizar la recuperación, sino que es consistente.