



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 287 418**

51 Int. Cl.:
H04N 7/24 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Número de solicitud europea: **03254401 .7**

86 Fecha de presentación : **11.07.2003**

87 Número de publicación de la solicitud: **1383335**

87 Fecha de publicación de la solicitud: **21.01.2004**

54 Título: **Método y sistema eficaz para entregar recursos en un medio de difusión.**

30 Prioridad: **12.07.2002 US 395654 P**

45 Fecha de publicación de la mención BOPI:
16.12.2007

45 Fecha de la publicación del folleto de la patente:
16.12.2007

73 Titular/es: **Ensequence, Inc.**
111 S.W. 5th Avenue, Suite 2200
Portland, Oregon 97204, US

72 Inventor/es: **Wallace, Michael W. y**
Westerman, Larry A.

74 Agente: **Elzaburu Márquez, Alberto**

ES 2 287 418 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Método y sistema eficaz para entregar recursos en un medio de difusión.

5 Este invento se refiere, en general, a sistemas de ordenador y a aplicaciones de software para ellos y, más específicamente, a la entrega de recursos de software.

10 Una corriente de datos de transmisión por un medio digital contiene una diversidad de tipos de contenido diferentes. Las transmisiones de radio digital por satélite, por poner un ejemplo, incluyen un contenido de audio digital para un número de canales diferentes, intercalado con un contenido de datos. El contenido de datos incluye información tal como títulos largos, nombres de artistas y otra información relacionada con el programa. Las transmisiones de televisión digital por satélite o por cable, por poner otro ejemplo, incluyen contenido de audio y de vídeo para una pluralidad de canales, intercalado con contenido de datos. El contenido de datos incluye calendarios de programación, datos de correlacionado de canales, y otros tipos de información. En ambos casos, un receptor/descodificador digital recibe y trata las transmisiones de datos para presentar audio, vídeo y otros datos al usuario.

15 Un receptor digital capaz de tratar transmisiones digitales, tal como un típico módulo de conexión multifunción ("STB") para televisión doméstica que trata señales digitales de televisión, permite la posibilidad de mejorar aún más la experiencia de un usuario al proporcionar contenidos adicionales. Al igual que se han transmitido programas de audio secundarios o con subtítulos ocultos junto con transmisiones de televisión analógica para mejorar el disfrute de la televisión por parte de los usuarios, también podrían transmitirse al STB tipos más variados de contenidos adicionales, a fin de ampliar la experiencia del espectador en un contexto de televisión digital. El STB constituye, al menos, un sistema de ordenador básico y, así, puede generar contenido adicional de audio y de vídeo a partir del contenido de datos intercalado en la corriente del contenido de audio y de vídeo. Por ejemplo, si el programa es un programa de noticias, podrían transmitirse al STB los últimos detalles u otros titulares en forma de bloques de datos que el STB puede descodificar y ofrecer al usuario. Similarmente, si el programa es un programa de deportes, podrían enviarse al STB los resultados, estadísticas o comentarios de audio, para descodificarlos para el espectador.

20 Además, debido a la potencia de cálculo inherente del STB, éste podría hacer posible que un usuario pudiera participar en televisión interactiva. Por ejemplo, los espectadores podrían participar en programas de juegos, en escrutinios en directo y realizar compras. EL STB puede servir como terminal para permitir que los usuarios interactúen con las emisiones de televisión en forma muy parecida a la que los ordenadores personales permiten que sus usuarios interactúen con Internet. Aplicaciones, mandos de control, imágenes, texto y otros datos pueden descargarse al ordenador a través de la corriente de datos para soportar y facilitar contenidos adicionales y programación interactiva.

25 La Figura 1 ilustra cómo podrían transmitirse contenidos adicionales y programación interactiva a un STB corriente conocido, a través del cual puede acceder un usuario. En un sitio de emisión 100, se introducen, en un multiplexador 108 de datos, material fuente 104 de audio y de vídeo junto con código 112 de aplicaciones, mandos 116, imágenes 120, datos de texto 124 y otros datos (no mostrados). El multiplexador 108 de datos multiplexa las diversas entradas para formar una corriente 130 de datos emitida, que es transmitida vía satélite y/o cable, a un sitio 140 de recepción. En el sitio 140 de recepción, la corriente 130 de datos emitidos es tratada y desmultiplexada por el STB 150 para formar material 154 de audio y vídeo, código 162 de aplicaciones, mandos 166, imágenes 170, datos de texto 174 y otros datos (no mostrados) recibidos. El material 154 de audio y de vídeo recibido, deseado, puede ser seleccionado por un usuario a través del STB 150 y aplicarse a través de un receptor 180 de televisión, y el usuario puede acceder al código 162 de aplicaciones, los mandos 166, las imágenes 170, los datos de texto 174 y los otros datos (no mostrados) a través del STB 150.

30 Se apreciará que la representación de la Figura 1 simplifica en cierto modo los procesos de emisión y recepción, sin mostrar, por tanto, algunos de los problemas a los que se enfrentan los diseñadores de programas, los ingenieros de emisión y otros participantes en ese campo. Una cuestión particular es la manera en que el material fuente 104 de audio y vídeo, el código 112 de aplicaciones, los mandos 116, las imágenes 120, los datos de texto 124 y otros datos (no mostrados), son realmente multiplexados en el sitio 100 de emisión mediante el multiplexador 108 de datos y son desmultiplexados y tratados por el STB 150 en el sitio 140 de recepción. La corriente 130 de datos emitidos está compuesta por muchos bloques discretos de información que, finalmente, son tratados por el STB 150 con el fin de permitir el acceso de los usuarios a los programas.

35 La Figura 2 muestra una corriente 200 de emisión de televisión digital representativa, corrientemente conocida en la técnica. La corriente emitida contiene una pluralidad de bloques de vídeo ("V") 210, bloques de audio ("A") 220 y bloques de datos ("D") 230. Cada uno de los bloques constituye parte de un paquete, y cada paquete está etiquetado de acuerdo con su tipo. Por ejemplo, los paquetes con el identificador de paquetes ("PID") 0, tienen un formato especial que describe la localización y/o el identificador de otros tipos de paquetes especiales. Otros valores de PID contienen otros tipos de datos, cuyos detalles se describen en una tabla de mapa de programas que proporciona una lista de los tipos de ficheros asociados con cada PID.

40 Los diferentes tipos de bloques son extraídos de la corriente de datos como se muestra y son concatenados para formar corrientes de datos coherentes, que pueden ser descodificadas para presentación al usuario. Típicamente, estos bloques son encaminados por un desmultiplexador (no mostrado) que reconoce los códigos que identifican los bloques 210 de vídeo, los bloques 220 de audio y los bloques 230 de datos, y encamina cada uno de ellos a una memoria

intermedia apropiada (no mostrada). Por ejemplo, los bloques 240 de vídeo relacionados son recogidos de la corriente 200 de datos. Estos bloques de vídeo son encaminados a una memoria intermedia (no mostrada) y son descodificados para presentarlos al espectador. Similarmente, los bloques 250 de audio identificados y los bloques 260 de datos identificados, son retirados de la corriente de datos y son encaminados a los medios de descodificación apropiados (no mostrados) para presentar su contenido al usuario. Se apreciará que la mayoría de los bloques son bloques 210 de vídeo porque, típicamente, los datos de vídeo consumen más almacenamiento de datos y más ancho de banda que otras formas de datos.

Dos aspectos de la ilustración que antecede de la corriente 200 de datos presentan problemas particulares. En primer lugar, los bloques 230 de datos se unen a la corriente 200 de datos emitida en un modo de “agárrese como pueda” en términos de ancho de banda disponible. Como se ha descrito previamente y como se apreciará, el ofrecer contenido de vídeo y de audio consume gran parte del ancho de banda disponible. La transmisión de un contenido adicional puede suponer la inserción de muchos bloques 230 de datos adicionales en la corriente 200 de datos transmitidos. Una transmisión típica de televisión digital utiliza una corriente de datos constituida por un gran número de pequeños bloques individuales. Estos bloques, denominados paquetes de corriente de transporte (“TS”), son bloques de tamaño fijo codificados y descodificados por capas de nivel inferior del protocolo de transmisión. Por ejemplo, en el caso del estándar del Grupo de Expertos en Películas para las transmisiones digitales de alta velocidad (MPEG-2), cada paquete de TS, ya contenga vídeo, audio u otros datos, tiene un tamaño de 188 bytes. Con un pequeño bloque constructivo con el que trabajar, cualquier contenido adicional, aún remotamente sofisticado, ocupa muchos bloques de datos para comunicar un directorio de ficheros disponibles y los propios ficheros necesarios para proporcionar contenido adicional.

En segundo lugar, podría consumirse una cantidad de almacenamiento relativamente grande si se guardasen todos los bloques entrantes en el STB. La grabación de programas completos se realiza utilizando cintas, discos duros o CD-ROM, que superan la capacidad de la memoria intermedia incluida en un STB 150 típico (Figura 1). Así, la memoria intermedia utilizada para soportar la ejecución continua de un programa seleccionado y cualquier contenido adicional relacionado, constituye un recurso relativamente precioso. Al igual que con cualquier otro recurso precioso, su gestión cuidadosa es un objetivo importante.

Una cuestión adicional relacionada con la transmisión de un contenido adicional es el hecho de que puede desearse que se reciba contenido adicional y se almacene para uso ulterior. Los bloques de audio y de vídeo son recogidos, almacenados en la memoria intermedia y presentados al usuario sustancialmente en forma de corriente en tiempo real. Así, si un usuario se engancha a un programa en ejecución, entra en el programa a partir de ese punto en adelante y no es necesario prever la captura de bloques de audio y de vídeo transmitidos antes de que el usuario accediese al programa. Por otro lado, los datos que incorporan contenido adicional, que va desde gráficos superpuestos hasta comentarios de audio, pueden extenderse por un gran número de bloques de datos intercalados situados entre los bloques de audio y de vídeo para todo el programa. Muchos bloques de datos necesarios para soportar un contenido adicional pueden haber sido transmitidos antes de que el usuario acceda al programa. No obstante, los bloques previamente transmitidos pueden ser solicitados en un momento posterior cuando se inicie el contenido adicional. Como tales datos están intercalados entre la programación de audio y de vídeo, una forma de asegurar la disponibilidad de todos los datos deseados para una aplicación con un contenido adicional particular, es transmitir repetidamente los datos en un bucle recurrente. De esta forma, incluso si un usuario activa su STB después de haberse enviado ya algunos bloques de datos deseados, los bloques de datos que faltan pueden conseguirse a partir de un bucle subsiguiente.

La Figura 3 ilustra una construcción de datos en bucle corrientemente conocida que, comúnmente, se denomina “carrusel de datos” 300. El carrusel de datos 300 está dibujado a partir de bloques de datos 230 aislados, recogidos (Figura 2), y trata la cuestión descrita en el párrafo anterior. Haciendo referencia a la Figura 3, por ejemplo, se supondrá que se requiere una cadena 310 del carrusel de datos para presentar una serie de mandos en pantalla. La presentación de estos mandos está soportada por una estructura de datos compuesta por un directorio 320 y tres ficheros, el fichero 1 330, el fichero 2 340 y el fichero 3 350, cada uno de los cuales está asociado con uno de la serie de mandos en pantalla. Así, con el fin de formar los mandos deseados, el STB utiliza toda la cadena 310. Por ejemplo, puede suponerse que un usuario activa el STB en un instante t^*360 que está en un punto intermedio durante una transmisión de bloques de datos que constituyen el fichero 1 330. Algunos de los bloques de datos que constituyen el fichero 1 330 fueron recibidos antes de t^*360 y, así, el fichero 1 330 no se guardará completo en la memoria del STB. Sin embargo, la cadena 310 de datos forma parte de un bucle recurrente 300 de construcciones de datos que están siendo transmitidos. En consecuencia, si algunos de los bloques de datos que forman el fichero 1 330 faltaba en una transmisión anterior, esos bloques de datos pueden ser adquiridos durante una repetición subsiguiente del carrusel 300 de datos, que es enviado en una revolución continua, en vista del nombre “carrusel” 300 de datos.

Otras dos cuestiones surgen del carrusel 300 de datos construido para transmitir bloques de datos. Primero, como se apreciará, muchos de los datos del carrusel 300 de datos, son retransmitidos una y otra vez. No hay necesidad de que el STB capture estos datos cada vez que son transmitidos y hacerlo supondría desperdiciar los recursos de tratamiento del STB.

Segundo, y quizás algo más que un problema, es la determinación de cómo asignar espacio de memoria intermedia en el STB para contenido adicional que incorpore datos. En la actualidad, hay, al menos, cuatro métodos para asignar espacio para los datos. En un primer método, se asigna una memoria intermedia de tamaño fijo para tales datos y los bloques de datos se dirigen a la memoria intermedia mediante el demultiplexador previamente descrito. Si el

ES 2 287 418 T3

volumen de bloques de datos es demasiado grande para que quepan en la memoria, el desmultiplexador inicia una petición de memoria intermedia adicional del soporte intermedio o de un sistema operativo que controle el hardware. Una vez concedida la memoria intermedia adicional, el desmultiplexador puede continuar, entonces, almacenando los bloques de datos. Sin embargo, la petición por parte del desmultiplexador de memorias intermedias de almacenamiento adicionales cada vez que se supera el espacio, puede ralentizar el proceso de desmultiplexado para grandes bloques de datos.

En un segundo método, se asigna más memoria de la que se espera necesitar para que el desmultiplexador almacene bloques de datos. El desmultiplexador puede, entonces, liberar la memoria no utilizada para otros usos dentro del STB. Este método evita el retardo del primer método al solicitar inmediatamente más espacio de memoria intermedia si las recepciones de datos iniciales superan el espacio asignado. Por otro lado, el segundo método requiere que se reserve una cantidad de memoria mayor que la esperada para los bloques de datos. Esto puede tener, como consecuencia, un desperdicio de valiosos recursos de hardware. La subsiguiente liberación de espacio de memoria intermedia no utilizado puede mejorar el derroche de asignación inicial. Sin embargo, una vez que el espacio de memoria intermedia alcanza el proceso negociado previamente descrito, este método adolece entonces de los potenciales problemas de retardo asociados con el primer método, tanto en lo que se refiere a la reserva como a la liberación de recursos de memoria, en la forma apropiada.

En un tercer método, las transmisiones de datos pueden limitarse a un tamaño fijo o, al menos, a incrementos de tamaño fijo. Este método reduce los retardos de tratamiento y de carga resultantes de que el desmultiplexador reserve y libere espacio de memoria intermedia según sea apropiado. Sin embargo, este tercer método limita la flexibilidad del sistema al restringir el tipo o la cantidad del contenido adicional para el que se utilizarán los datos. Además, en honor de las limitaciones de tamaño fijas, las aplicaciones pueden utilizar de manera conservadora los recursos de hardware asignados. Así, la memoria asignada para bloques de datos puede quedar sin uso o infrautilizarse, y parte de ese espacio de memoria se desperdiciaría.

El cuarto método es un híbrido de algunos de los métodos anteriores. Una vez iniciado el sistema, las memorias intermedias son asignadas inicialmente a un tamaño de datos determinado por las necesidades de almacenamiento de los conjuntos de datos iniciales recibidos. El tamaño de memoria intermedia asignado es entonces protegido en la cantidad especificada por los conjuntos de datos iniciales. Mientras que los conjuntos menores de datos pueden ser recibidos, los conjuntos mayores de datos serán rechazados. Este método tiene la ventaja de que aprovecha cierto uso representativo real del espacio de memoria intermedia para establecer el tamaño de memoria intermedia asignado, en lugar de utilizar hipotéticos tamaños de ficheros. Sin embargo, este método tiene, también, desventajas. Por ejemplo, si el tamaño de memoria intermedia asignado inicialmente es grande, se desperdiciará subsiguientemente mucha de esa memoria intermedia. Por otro lado, si el tamaño asignado de la memoria intermedia es demasiado pequeño, entonces la flexibilidad del sistema para utilizar conjuntos de datos mayores, se ve restringida basándose en lo que se suponía era un tamaño inicial límite de conjunto de datos.

Más allá de estos métodos, los sistemas podrían intentar enfoques más flexibles para la asignación de espacio de memoria intermedia. Por ejemplo, utilizando el protocolo DSM-CC, definido como parte del estándar internacional ISO-IEC 13818-6, una entrada de directorio para un fichero contiene una designación de tamaño del fichero. Esta designación podría ser utilizada por el desmultiplexador para iniciar peticiones de reasignación antes de recibir el fichero. Sin embargo, el tamaño de fichero especificado en esta norma representa el tamaño de los datos comprimidos en la corriente emitida. Dado que los datos han de ser descomprimidos para utilizarlos, y el tamaño de los ficheros comprimidos varía mucho dependiendo de la naturaleza de los datos particulares que se comprimen, el tamaño de fichero especificado puede ser una mala indicación de, incluso, el tamaño relativo de los ficheros de datos descomprimidos.

La ETSI TS 10812 V1.2.2 proporciona una especificación técnica para la emisión de vídeo digital para una plataforma doméstica multimedia en la que un carrusel de datos incluye un descriptor de ID de carrusel como fichero MPEG que, como se ha establecido en lo que antecede, es un bloque de tamaño fijo y relativamente pequeño.

Otra exposición pertinente puede verse a partir de

- (1) www.rechnet.com/dab;
- (2) http://www.ibiquity.com/technology/documents/SY_TN_5032_000.pdf;
- (3) http://webapp.etsi.org/workprogram/Report_WorkItem.asp?KI_ID=19901;
- (4) www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=31537&ICSI=35&ICS2=40&ICS3=;
- (5) iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=25039&ICSI=35&ICS2=40&ICS3=; y
- (6) www.opentv.com.

ES 2 287 418 T3

Así, en la técnica existe una necesidad insatisfecha para reconocer de manera eficaz ficheros modificados en un carrusel de datos y para asignar de manera eficaz espacio de memoria intermedia para bloques de datos que están siendo leídos a una memoria de STB.

5 Las realizaciones del presente invento permiten el uso eficaz de los recursos de un módulo de conexión multi-función (STB) u otro receptor utilizado para tratar una corriente de datos emitidos. El invento está definido por las reivindicaciones 1 y 9.

10 Así, cuando se detecta el fichero de índice que lleva el indicador de cambio, el receptor de la corriente de datos busca el fichero que ha sido cambiado y reemplaza el fichero de datos desechado por el fichero de datos cambiado.

15 De acuerdo con otros aspectos del invento, un receptor que detecta el indicador de cambio del fichero de índice busca el fichero cambiado a partir de la corriente de datos y reemplaza un fichero de datos desechado por el fichero de datos cambiado. Además, el fichero de índice incluye un indicador de versión del fichero de índice, que indica una versión del fichero de índice y, así, indica cuando se ha cambiado el fichero de índice. Asimismo, el fichero de índice incluyen indicadores de versión de fichero para los ficheros de la serie de ficheros con que está asociado el fichero de índice. Los indicadores de versión de fichero señalan cuando se ha cambiado un fichero desde la recepción de un bloque de índice previo. Asimismo, el fichero de índice incluye un indicador de tamaño del fichero que permite determinar qué fichero o ficheros se han cambiado y/o el tamaño de una o más memorias intermedias que han de asignarse para recibir el o los ficheros cambiados.

En lo que sigue se describen con detalle las realizaciones preferidas y las alternativas del presente invento, sólo a modo de ejemplo, con referencia a los siguientes dibujos, en los que:

25 la Figura 1 es un diagrama de bloques de la composición, transmisión y tratamiento de una corriente de datos utilizada en una transmisión de televisión digital de la técnica anterior;

la Figura 2 es un diagrama de bloques de una corriente de datos de vídeo, audio y otros bloques de datos utilizados en una transmisión de un medio digital, usual, de la técnica anterior;

30 la Figura 3 es un diagrama de bloques de un carrusel de datos utilizado en una transmisión de un medio digital usual, de la técnica anterior;

35 la Figura 4 es un diagrama de bloques de un carrusel de datos que incorpora un fichero de índice de acuerdo con una realización del presente invento;

las Figuras 5A y 5B son diagramas de bloques de un fichero de índice de acuerdo con una realización del presente invento;

40 la Figura 6 es un diagrama de bloques para crear y actualizar un fichero de índice en un sitio de transmisión de acuerdo con una realización del presente invento;

la Figura 7 es un diagrama de bloques para recibir y usar un fichero de índice en un sitio de recepción de acuerdo con una realización del presente invento;

45 la figura 8 es un diagrama de bloques de un sistema de transmisión con control del medio/tratamiento de datos que utiliza un fichero de índice de una realización del presente invento; y

50 la Figura 9 es un diagrama de bloques de un sistema de recepción con control del medio/tratamiento de datos que utiliza un fichero de índice de una realización del presente invento.

A modo de síntesis, las realizaciones del presente invento proporcionan un método, un medio legible por ordenador y un sistema de datos para indicar cuando un fichero de datos desechado de una serie de ficheros de datos, ha sido reemplazado por un fichero de datos modificado. Mediante el uso de las realizaciones del presente invento, se crea un fichero de índice y se inserta en la serie de ficheros de datos. El fichero de índice incluye un indicador de cambio que señala que el fichero de datos modificado ha reemplazado al fichero desechado. El fichero de índice es transmitido repetidamente con la serie de ficheros de datos.

60 La Figura 4 muestra un carrusel 400 de datos que incorpora un fichero 415 de índice de acuerdo con una realización del presente invento. Para simplificar la comparación con el carrusel 300 de datos de la técnica anterior (Figura 3), el carrusel 400 de datos incorpora algunos de los elementos del carrusel 300 de datos (Figura 3), incluyendo un directorio 320 y tres ficheros; fichero 1 330, fichero 2 340 y fichero 3 350. Cada uno de estos elementos comunes es numerado de forma idéntica que en la Figura 3, para resaltar que estos elementos no han de cambiarse de acuerdo con el presente invento. Se apreciará que un cambio notable entre el carrusel 300 de datos de la técnica anterior (Figura 3) y el carrusel 65 400 de datos que incorpora una realización del presente invento, es la adición del fichero 415 de índice al carrusel 400 de datos. El fichero 415 de índice puede abarcar múltiples bloques de datos. Se apreciará que esta amplitud es similar a los otros elementos tales como el directorio 320 y los tres ficheros, fichero 1 330, fichero 2 340 y fichero 3 350. Al igual que ocurre con otros elementos, el fichero 415 de índice es transmitido de forma recurrente.

ES 2 287 418 T3

Algunos de los principios de funcionamiento del carrusel 400 de datos recuerdan a los del carrusel 300 de datos de la técnica anterior (Figura 3). Por ejemplo, los bloques 230 de datos (Figura 2) son transmitidos recurrentemente en la corriente 200 de datos emitida, y son recogidos y almacenados temporalmente para su uso. Se apreciará que un sistema que utilice el carrusel 400 de datos reacciona de forma diferente a la transmisión recurrente de los elementos del carrusel 400 de datos. En un sistema que utilice el carrusel 400 de datos de acuerdo con una realización del presente invento, una vez que se ha accedido al sistema y se han almacenado temporalmente un conjunto de ficheros 330, 340 y 350 contenidos en el directorio 320 apropiados para las selecciones del usuario corriente, el sistema no cargará ni almacenará, en forma adecuada, ficheros adicionales hasta que se reciba un fichero 415 de índice que indique qué contenidos del fichero de índice y, por tanto, de otros ficheros del carrusel 400 de datos, han cambiado. Esto contrasta con un carrusel de datos de la técnica anterior, en el que los ficheros podrían ser leídos y almacenados temporalmente de modo continuo con cada iteración, en caso de que hubiese cambiado cualquiera de los ficheros.

En consecuencia, el sistema que utilizase una realización actualmente preferida del presente invento, economizaría los recursos y/o el tiempo requerido para leer y almacenar temporalmente ficheros idénticos a sus predecesores. En cambio, el presente invento leería y almacenaría temporalmente, de manera ventajosa, ficheros nuevos sólo cuando hubiesen tenido lugar uno o más cambios en los ficheros. En una realización actualmente preferida, solamente los ficheros que realmente han cambiado son leídos y almacenados temporalmente, como se describirá con mayor detalle en lo que sigue, economizando así aún más recursos y/o tiempo. Sin embargo, sigue estando de acuerdo con los principios generales del presente invento si múltiples ficheros, o todos los ficheros, del carrusel de datos, son leídos y almacenados temporalmente cada vez que se realiza un cambio en el fichero 415 de índice, para significar que se ha producido, al menos, un cambio en uno o más de los ficheros del carrusel 400 de datos.

La Figura 5A muestra una vista más detallada de un fichero 500 de índice ilustrativo de acuerdo con el presente invento. En una realización actualmente preferida, el fichero 500 de índice tiene cuatro partes: un campo 504 de versión del fichero índice; una lista 508 de ficheros; una lista 512 de versiones de ficheros y una lista 516 de tamaños de ficheros. El campo 504 de versión del fichero índice es actualizado cada vez que se cambia el fichero 500 de índice respecto de una versión previa. La lista 508 de ficheros enumera los otros ficheros del carrusel de datos asociados con el fichero 500 de índice. La lista 512 de versiones de ficheros enumera un identificador de la versión corriente de cada uno de los ficheros enumerados en la lista 508 de ficheros. La lista 516 de tamaños de ficheros enumera el tamaño corriente de cada fichero contenido en la lista 508 de ficheros. En una realización actualmente preferida, existe una correspondencia de uno a uno entre la lista 508 de ficheros, la lista 512 de versiones de ficheros y las listas 512 de tamaños de ficheros. Por tanto, las listas 508, 512 y 516 pueden concebirse, lógicamente, como una tabla con valores en cada columna de cada fila para asociar los valores apropiados con los ficheros apropiados.

El campo 504 de versión de fichero de índice indica cuando se ha cambiado el fichero 500 de índice, señalando por tanto cuando se han cambiado uno o más de los ficheros del carrusel de datos representado por el fichero 500 de índice. El valor 506 de versión del fichero de índice del campo 504 de versión del fichero de índice puede significar un cambio del fichero 500 de índice de diversas formas. Un valor 506 de versión de fichero de índice almacenado como versión 504 de fichero de índice es incrementado adecuadamente, cambiado en uno y otro sentido entre un conjunto limitado de valores establecidos, o modificado de otro modo. Como se apreciará, es el cambio del valor 506 de versión del fichero de índice almacenado en el campo 504 de versión del fichero de índice, respecto de un identificador de versión previo a un identificador de versión siguiente lo que tiene importancia y no a qué cambia el valor 506 de versión del fichero de índice.

El resto de los elementos del fichero 500 de índice crea lógicamente una tabla que enumera los atributos corrientes acerca de cada uno de los ficheros 510 contenidos en la lista 508 de ficheros. Por cada fichero 510 contenido en la lista 508 de ficheros, un valor 514 de versión de fichero corriente está contenido en la lista 512 de versión de fichero. Como se ha descrito previamente en relación con el valor 506 de versión del fichero de índice almacenado en el campo 504 de versión del fichero de índice, los valores 514 de versión de fichero utilizados en la lista 512 de versiones de fichero podría, de diversas formas, significar un cambio. Un valor 514 de versión de fichero almacenado en la lista 512 de versiones de fichero podría ser incrementado, cambiado en vaivén entre un conjunto limitado de valores establecidos, o modificado de otro modo. Como se apreciará, lo que importa es el cambio del valor 514 de versión de fichero de la lista 512 de versiones de fichero, respecto de un identificador de versión previa a un identificador de siguiente versión, y no el valor específico al que se cambia el valor 514 de versión de fichero. Asimismo, para cada uno de los ficheros 510 contenidos en la lista 512 de versiones de fichero, un tamaño 518 de fichero está contenido en la lista 516 de tamaños de fichero. El tamaño 518 de fichero contenido en la lista 516 de tamaños de ficheros se utiliza para asignar espacio de memoria intermedia para ficheros cambiados, como se describirá más adelante.

Como se ha mencionado previamente, el fichero 500 de índice es transmitido de manera recurrente con los otros ficheros 510 del carrusel de datos con el cual está asociado el fichero 500 de índice. Si no ha cambiado ninguno de los ficheros 510, el fichero 500 de índice permanece sin cambios y, ventajosamente, los ficheros 510 del carrusel de datos no se cargan ni se almacenan temporalmente. Es cuando se cambia el fichero 500 de índice para reflejar un cambio que tiene lugar en uno de los ficheros 510, cuando el sistema que hace uso de una realización del presente invento actúa sobre la información contenida en el fichero 500 de índice.

El funcionamiento y el efecto del fichero 500 de índice puede ilustrarse describiendo el resultado que se obtiene cuando se sustituye el fichero 500 de índice por un fichero 550 de índice actualizado, como se muestra en la Figura 5B. La Figura 5A muestra como es el fichero 500 de índice para una serie de ficheros: fichero 1 v.1 520, fichero 2 v.6 534,

ES 2 287 418 T3

fichero 3 v.3 538 y fichero 4 v.17 542. Como también se muestra en la Figura 5A, el fichero 500 de índice contiene información corriente acerca de cada uno de los ficheros 530, 534, 538 y 542. Específicamente, el fichero de índice enumera los nombres de los ficheros en la lista 508 de ficheros, la versión de cada uno de los ficheros en la lista 512 de versiones de ficheros, y el tamaño de cada uno de los ficheros en la lista 516 de tamaños de ficheros. Asimismo, el fichero de índice contiene su valor 506 de versión corriente como “21” en el campo 504 de versión de fichero de índice. Si, en este ejemplo, se incrementa el campo 504 de versión de fichero de índice en un valor entero de uno cada ve que se cambie partiendo de un valor de cero, el fichero 500 de índice ha cambiado 21 veces para reflejar un número correspondiente de casos en los que se cambiaron los ficheros asociados 530, 534, 538 y 542.

La Figura 5B ilustra un fichero 550 de índice actualizado para la misma serie de ficheros 530, 534, 538 y 542 de los que uno de los ficheros, el fichero 3 v.3 538 (Figura 5A), con un tamaño de fichero de 10934, ha sido substituido por el fichero 3 v.4 580 con un tamaño de fichero de 11385. Como el fichero 3 v.3 538 (Figura 5A) ha sido reemplazado por el fichero 3 v.4 580 (Figura 5B), se introduce un fichero 550 de índice actualizado en el carrusel de datos que contiene estos ficheros. Esto le indica al sistema que hace uso de una realización del presente invento, que se ha cambiado un fichero del carrusel.

Como el cambio entre los carruseles representados por los ficheros de índice 500 y 550 (Figuras 5A y 5B, respectivamente) supone un cambio de versión del fichero 3, de fichero 3 v3 538 con un tamaño de 10934 (Figura 5A) a fichero 3 v.4 580 con un tamaño de 11385 (Figura 5B), se realizan tres cambios para actualizar el fichero 500 de índice a fin de crear el fichero 550 de índice actualizado. En primer lugar, se cambia el valor 507 de versión de fichero de índice de “21” al valor 584 de “22” de versión del fichero de índice actualizado. En segundo lugar, en la lista 512 de versión de ficheros, el valor 514 de versión de fichero de “3” para el fichero 3 515, se cambia a su valor 586 de versión actualizado de “4”, para el fichero 3. Finalmente, en la lista 516 de tamaños de ficheros, el valor 518 de tamaño de fichero de “10934” 519 para el fichero 3 se cambia al valor 588 de tamaño de fichero actualizado de “11385” para el fichero 3. Con esta información en el fichero 500 de índice actualizado (Figura 5B), el sistema que hace uso de una realización del presente invento puede ser informado, ventajosamente, de cambios en uno o más ficheros del carrusel de datos sin tener que cargar y almacenar temporalmente, de manera constante, ficheros de datos no cambiados, con lo que se ahorran tiempo y recursos.

Continuando con este ejemplo, puede suponerse que el sistema ya ha recibido un conjunto de ficheros de datos 530, 534, 538 y 542, representados por el fichero 500 de índice. El sistema explora el carrusel buscando un fichero 550 de índice actualizado investigando los paquetes que llevan una PID de fichero de índice, como se ha descrito previamente. Al encontrar un paquete con una PID de fichero de índice, el sistema comprueba el paquete para determinar si el valor 506 de versión de fichero de índice ha cambiado desde que se recibió el último fichero de índice. Si el valor 506 de versión de fichero de índice es igual que el de la última versión de fichero de índice recibida, el sistema no toma ninguna medida. Por otro lado, si el valor 506 de versión del fichero de índice ha cambiado respecto de un valor previo, el sistema lee el fichero 550 de índice actualizado como primer paso para asegurar los ficheros actualizados del carrusel de datos.

Después de cargar el fichero de índice actualizado, en una realización preferida en la actualidad, el siguiente paso consiste en comprobar la lista 512 de versiones de fichero para determinar cual de los ficheros ha sido actualizado desde que se recibió el fichero de índice previo. Específicamente, se comprueban los valores 514 de versión de fichero para ver qué ha cambiado. En este caso, como la versión de fichero del fichero 3 ha cambiado pasando a ser la versión “4” 586, mientras que los otros valores de versión de fichero siguen siendo los mismos, se ha identificado el fichero cambiado. Habiendo identificado el fichero cambiado, se comprueba la lista 516 de tamaños de fichero en lo que respecta al fichero 3 580 y se lee el tamaño 588 del fichero cambiado “11385”. Una vez encontrado el tamaño del fichero cambiado, el sistema asigna entonces una memoria intermedia (no mostrada) capaz de acomodar el tamaño 588 de fichero cambiado “11385”. El fichero 3 580 cambiado es leído entonces incorporándolo a esta memoria intermedia. Se libera luego la memoria intermedia que contiene el fichero 3 538 previo, substituido, de forma que el espacio de memoria consumido por esa memoria intermedia pueda utilizarse para ficheros recibidos subsiguientemente o con otros fines.

Las Figuras 6 y 7 describen en forma de gráfica de flujo rutinas para generar y utilizar ficheros de índice, respectivamente. Haciendo referencia a la Figura 6, una rutina para generar tablas de índice comienza en un bloque 610. En un bloque 620 se crea un fichero de índice original registrando un número de ficheros del carrusel pertinente, cada uno de los cuales puede ser cambiado durante la emisión; un número de versión original para cada fichero; y un tamaño original para cada fichero. Los datos recogidos se registran en listas apropiadas, como se ha descrito en relación con las Figuras 5A y 5B. En un bloque 630, se introduce el fichero de índice en el carrusel de datos y se transmite junto con los ficheros con que está asociado el fichero de índice.

En el bloque de decisión 640, se determina si se ha realizado algún cambio a los ficheros asociados con el fichero de índice. Si no ha sido así, la rutina pasa por el bloque de decisión 640 hasta que se cambie uno de los ficheros asociados. Cuando se cambie un fichero asociado, en un bloque 650 se introduce en el carrusel el fichero modificado. En un bloque 660 se registra en el fichero de índice un tamaño de fichero modificado para el fichero cambiado. En un bloque 670, se cambian la versión de fichero para el fichero y una versión de fichero de índice para el fichero de índice. En el bloque 680, se introduce el fichero de índice revisado en el carrusel de datos, en lugar de la versión previa del fichero de índice, y se le transmite. La rutina continúa en el bloque de decisión 640, comprobando si los ficheros del carrusel han cambiado.

ES 2 287 418 T3

Haciendo referencia a la Figura 7, una rutina para utilizar la tabla de índice comienza en el bloque 710. En el bloque 720 se comprueba una versión de fichero de índice del fichero de índice recibido en el carrusel. En el bloque de decisión 730 se determina si las versiones de fichero de índice han cambiado desde que se revisó el fichero de índice previo. Se apreciará que, en lugar de comprobar una versión del fichero de índice, un sistema podría comprobar individualmente las versiones de fichero contenidas en el fichero de índice o comprobar los tamaños de fichero enumerados para ver si han cambiado. En cualquier caso, si las indicaciones del fichero de índice muestran que éste no ha cambiado, la rutina sigue a través del bloque de decisión 730 hasta que haya cambiado la versión del fichero de índice. Cuando la versión del fichero de índice ha cambiado, en el bloque 740 se identifica el fichero que ha cambiado. En una realización actualmente preferida, se determina si un fichero ha cambiado comprobando un número de versión de fichero. Alternativamente, también se podría determinar si ha cambiado un fichero comparando un tamaño de fichero para determinar si se ha modificado respecto del fichero de índice precedente.

Una vez que se han identificado el o los ficheros que han cambiado, en el bloque 750 se lee el tamaño del fichero modificado a partir de la tabla de índice, con independencia de si el fichero modificado es del mismo tamaño, de mayor tamaño o menor. En el bloque 770, el fichero modificado es leído a la memoria intermedia recién asignada. En el bloque 780, se libera la memoria intermedia que contiene la versión previa sustituida del fichero, de modo que el espacio de memoria utilizado pueda ser dedicado a otros usos. La rutina continúa comprobando, en el bloque de decisión 730 si se ha cambiado la versión del fichero de índice recibido.

La Figura 8 muestra un sistema de ordenador configurado como sistema 800 de transmisión de control de medios/tratamiento de datos, que puede ser hecho funcionar incorporando realizaciones del presente invento. El sistema 800 de ordenador es hecho funcionar controlando un dispositivo de presentación 802, tal como un televisor, y un subsistema de audio 804, tal como un sistema estéreo o de altavoces para vigilar la transmisión de audio, vídeo y transmisiones de datos. El sistema 800 transmite a través de un sistema 806 de difusión. El sistema 800 es controlado, adecuadamente, mediante entradas de usuario realizadas a través de un teclado 808 de usuario, cableado o inalámbrico.

El sistema 800 transmite a través del sistema de difusión 806 mediante un controlador 810 de entrada/salida, que dirige las señales hacia y desde un controlador de vídeo 812 y un controlador de audio 814 y una unidad central de tratamiento ("CPU") 816. El controlador 810 de entrada/salida es, adecuadamente, un multiplexador para encaminar bloques de datos de vídeo, bloques de datos de audio y otros bloques de datos encaminados a través de y/o generados por el sistema 800, a una CPU 816 para tratamiento y multiplexado. La CPU 816 comunica a través de un controlador 818 de sistema con dispositivos de entrada y de almacenamiento tales como una memoria de sólo lectura ("ROM") 820, una memoria 822 de sistema, un almacenamiento 824 de sistema y un controlador 826 de dispositivos de entrada. El sistema 800 ilustrado en la Figura 8 puede generar y/o transmitir, así, bloques de datos a través del sistema de difusión. Se apreciará que la CPU 816 y los componentes asociados, pueden ser hechos funcionar para formatear y generar los ficheros de índice de acuerdo con el presente invento

La Figura 9 muestra un sistema de ordenador configurado como un sistema 900 de recepción reconrol de medios/tratamiento de datos, que puede ser hecho funcionar para incorporar realizaciones del presente invento. El sistema 900, en lugar de transmitir datos a través de un sistema de difusión 806 (Figura 8), recibe datos de una red 906 tal como una red de cable digital de banda ancha, una red de satélite digital u otra red de datos. En vez de generar y multiplexar datos, el sistema 900 recibe un contenido de audio, vídeo y datos desde la red 906. El sistema 900 es, adecuadamente, un STB para controlar un dispositivo de presentación 902, tal como una televisión y un subsistema 904 de audio, tal como un sistema de altavoces o de estéreo. El sistema 900 recibe, también, entradas de usuario desde un teclado 908 de usuario, cableado o inalámbrico, cuya naturaleza puede ser la de un STB remoto.

El sistema 900 recibe una entrada desde la red 906 a través de un controlador 910 de entrada/salida, que dirige señales hacia y desde un controlador 912 de vídeo, un controlador 914 de audio y una unidad central de tratamiento (CPU) 916. En el caso de un STB, el controlador 910 de entrada/salida es, adecuadamente, un desmultiplexador para encaminar los bloques de datos de vídeo recibidos desde la red 906 a un controlador de vídeo 912 cuya naturaleza es la de un descodificador de vídeo, los bloques de datos de audio a un controlador 914 de audio, cuya naturaleza es la de un descodificador de audio, y para encaminar otros bloques de datos a una CPU 916 para tratamiento. A su vez, la CPU 916 comunica, a través de un controlador 918 de sistema, con dispositivos de entrada y de almacenamiento tales como una ROM 920, una memoria 922 de sistema, un almacenamiento 924 de sistema y un controlador 926 de dispositivos de entrada.

El sistema 900 ilustrado en la Figura 9 puede recibir, así, ficheros de datos entrantes de diversas clases, incluyendo los ficheros de índice de acuerdo con el presente invento. El sistema 900 puede reaccionar a los ficheros de índice para recibir y tratar ficheros de datos cambiados recibidos desde la red 906.

Aunque se ha ilustrado y descrito la realización preferida del invento, como se ha señalado anteriormente, pueden realizarse en ella muchos cambios sin salirse del ámbito del invento. En consecuencia, el alcance del invento no está limitado por la exposición de la realización preferida. Por el contrario, el invento debe determinarse enteramente por referencia a las reivindicaciones que siguen.

REIVINDICACIONES

5 1. Un método para indicar cuando un fichero de datos a sustituir de una serie de ficheros de datos, ha sido sustituido por un fichero de datos cambiado, cuyo método comprende:

reemplazar (650) un fichero de datos a sustituir (500) de una serie de ficheros de datos (320, 330, 340, 350) previstos en un carrusel (300) de emisión de datos, por un fichero de datos cambiado (550);

10 crear (670) un fichero (415) de índice, incluyendo el fichero (415) de índice un indicador de cambio que señala que el fichero de datos cambiado (550) ha reemplazado al fichero de datos a sustituir (500);

insertar (680) el fichero de índice en el carrusel de emisión de datos; y

15 transmitir repetidamente el fichero (415) de índice con la serie de ficheros de datos por cada revolución del carrusel de emisión de datos.

2. El método de la reivindicación 1, que comprende, además, mantener el indicador de cambio del fichero (415) de índice, incluyendo dicho mantenimiento del indicador de cambio:

20 mantener un indicador de versión en el fichero (415) de índice;

cambiar el indicador de versión del fichero (415) de índice cuando el fichero de datos a sustituir (500) ha sido reemplazado por el fichero de datos cambiado; e

25 insertar un tamaño (558) de fichero cambiado para el fichero de datos cambiado (580) en el fichero de índice.

3. El método de la reivindicación 2, en el que el indicado de versión se cambia incrementándolo cuando el fichero de datos a sustituir (500) ha sido reemplazado por el fichero de datos cambiado (550).

30 4. El método de la reivindicación 2 o la reivindicación 3, en el que el indicador de versión incluye un indicador (506) de versión del fichero de índice que señala la versión del fichero (415) de índice.

35 5. El método de la reivindicación 4, en el que el indicador (506) de versión del fichero de índice se cambia cuando se ha cambiado cualquiera de una serie de ficheros de datos.

6. El método de las reivindicaciones 2, 3, 4 o 5, en el que el indicador de versión incluye un indicador (512) de versión de fichero para indicar una versión del fichero (550) de datos cambiado.

40 7. Un medio legible por ordenador que comprende medios de código de programa de ordenador que, cuando son ejecutados por un ordenador, indican cuándo un fichero de datos a sustituir (500) de una serie de ficheros de datos, ha sido reemplazado por un fichero de datos cambiado (550) de acuerdo con el método definido en una cualquiera de las reivindicaciones 1 a 6.

45 8. Un ordenador cargado con los medios de código de programa de ordenador definidos en la reivindicación 7.

9. Un sistema de control de transmisión de medio para indicar cuando un fichero de datos a sustituir de una serie de ficheros de datos, ha sido reemplazado por un fichero de datos cambiado, cuyo sistema comprende:

50 un procesador que incluye:

un primer componente configurado para reemplazar un fichero de datos a sustituir (500) por un fichero de datos cambiado en una serie de ficheros de datos previstos en un carrusel (300) de emisión de datos; y

55 un segundo componente configurado para crear un fichero (415) de índice, generando el segundo componente un indicador de cambio que señala que el fichero de datos cambiado ha reemplazado al fichero de datos desechado (500);

60 estando configurado el segundo componente para insertar el fichero de índice en el carrusel (300) de emisión de datos; y

65 un transmisor acoplado al procesador, estando configurado el transmisor para transmitir repetidamente el fichero (415) de índice con la serie de ficheros de datos por cada revolución del carrusel de emisión de datos.

ES 2 287 418 T3

10. El sistema de la reivindicación 9, en el que el indicador de cambio del fichero (415) de índice, incluye:

un indicador de versión, siendo cambiado el indicador de versión cuando el ficheros de datos a sustituir (500) ha sido reemplazado por el fichero de datos cambiado (550); y

5

un tamaño de fichero cambiado para el fichero de datos cambiado en el fichero (415) de índice.

11. El sistema de la reivindicación 10 o la reivindicación 10, en el que el segundo componente está configurado, además, para incrementar el indicador de versión cuando el fichero de datos a sustituir (500) ha sido reemplazado por el fichero de datos cambiado (550).

10

12. El sistema de la reivindicación 10 o la reivindicación 11, en el que el indicador de versión incluye un indicador de versión de fichero de índice que señala la versión del fichero de índice.

13. El sistema de la reivindicación 12, en el que el segundo componente está configurado, además, para cambiar el indicador de versión del fichero de índice cuando se ha cambiado cualquiera de una serie de ficheros de datos.

15

14. El sistema de una cualquiera de las reivindicaciones 9 a 13, en el que el indicador de versión incluye un indicador de versión de fichero para señalar una versión del fichero de datos cambiado.

20

25

30

35

40

45

50

55

60

65

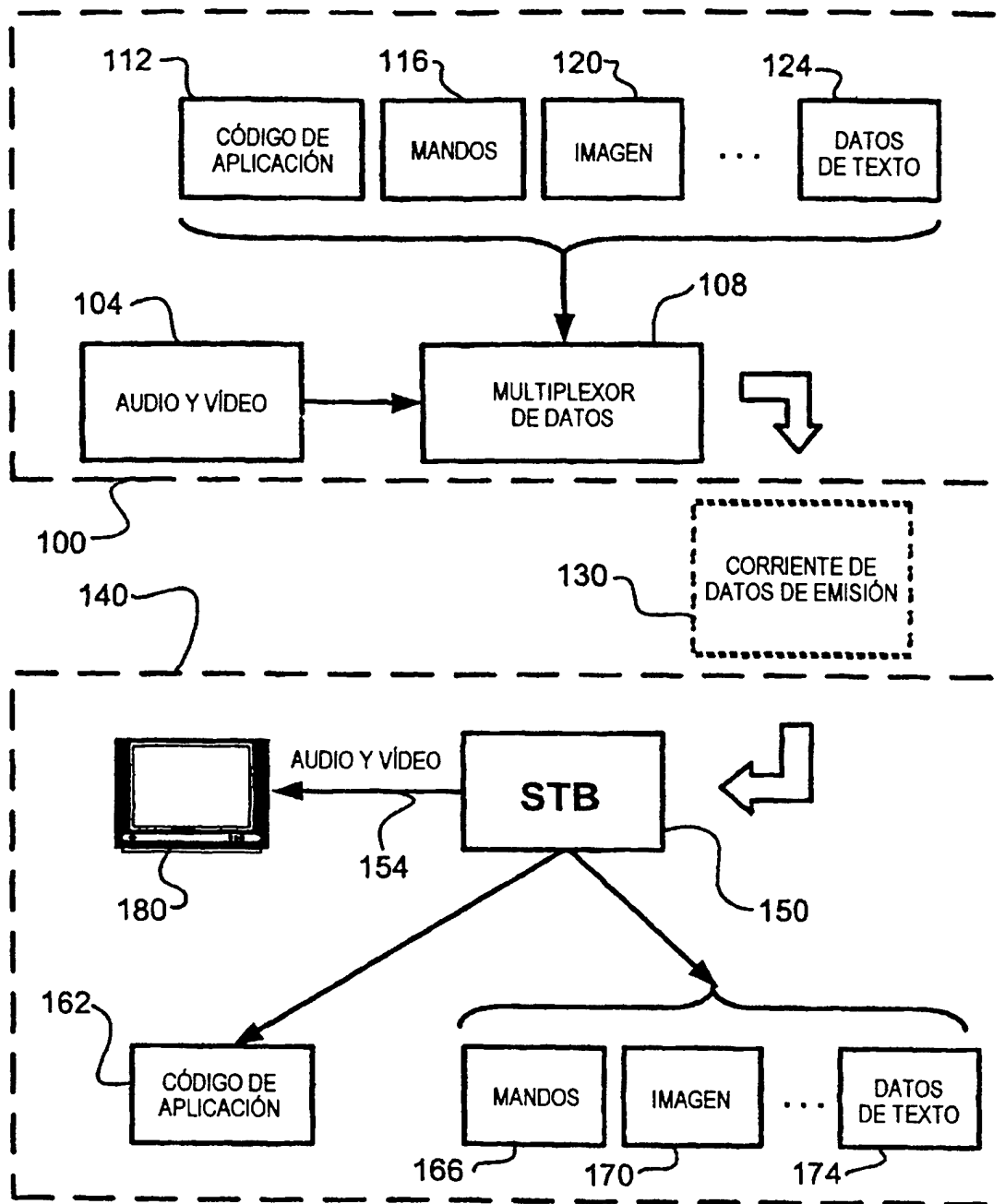


Fig. 1
(TÉCNICA ANTERIOR)

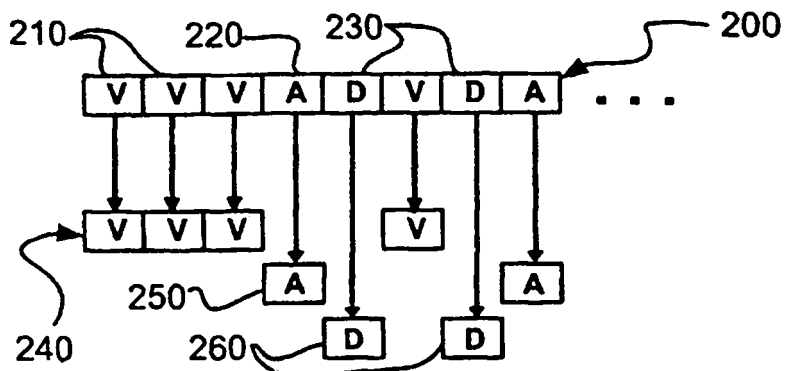


Fig. 2
(TÉCNICA ANTERIOR)

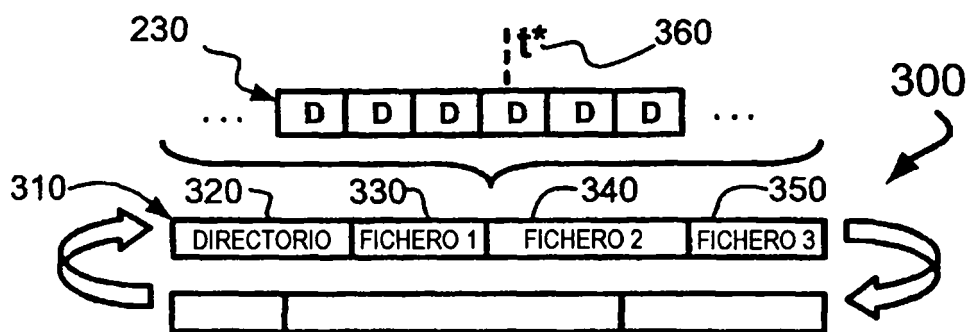


Fig. 3
(TÉCNICA ANTERIOR)

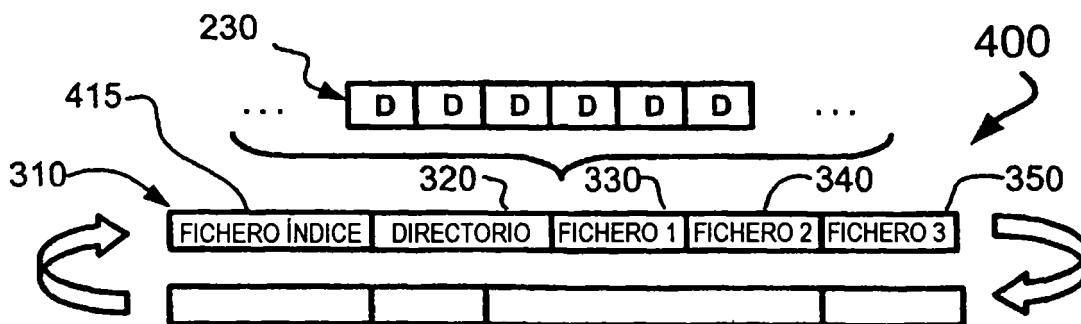


Fig. 4

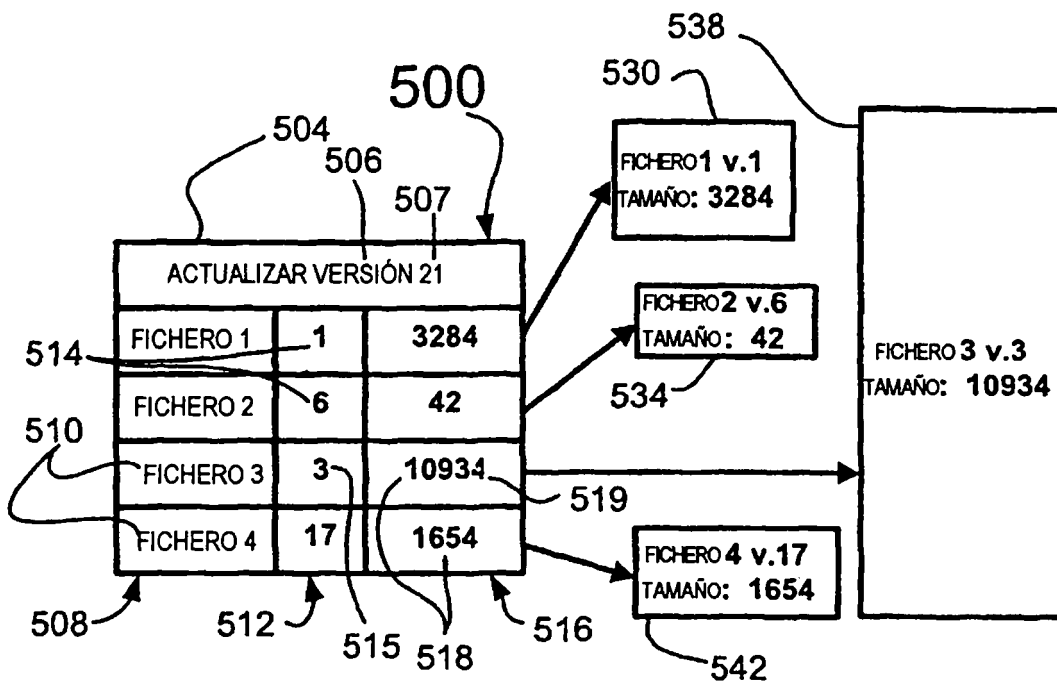


Fig. 5A

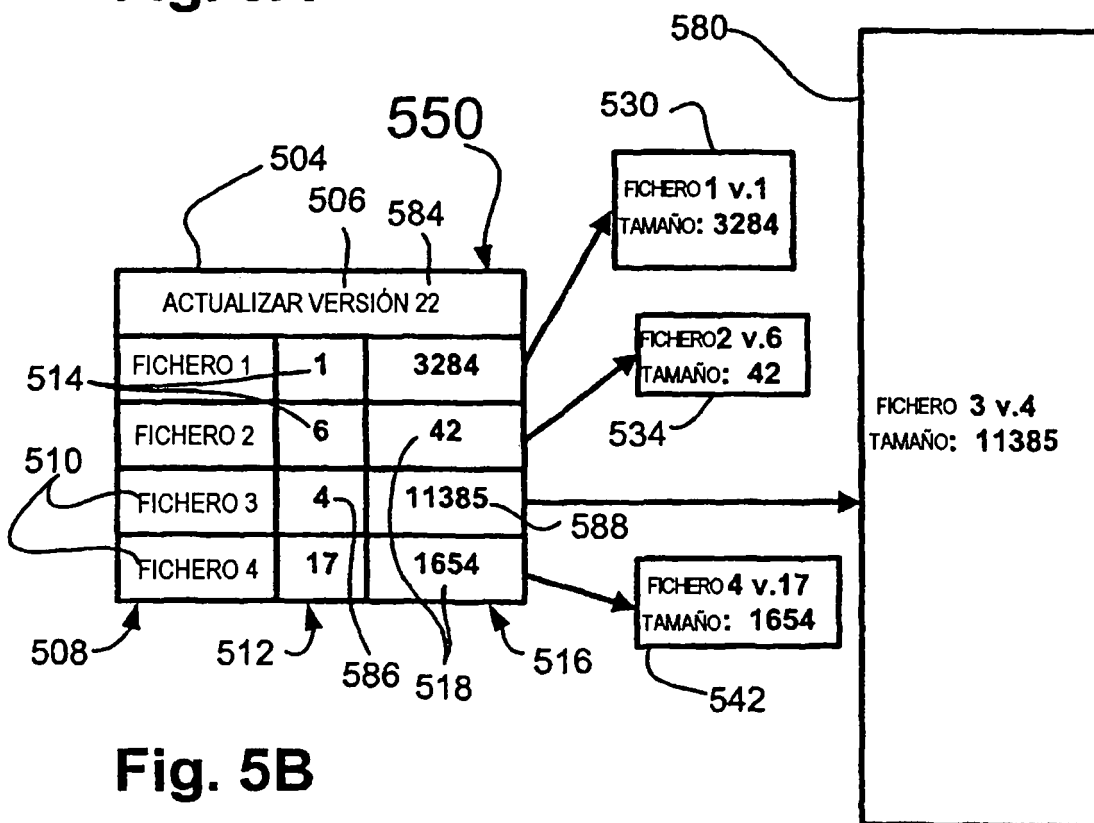


Fig. 5B

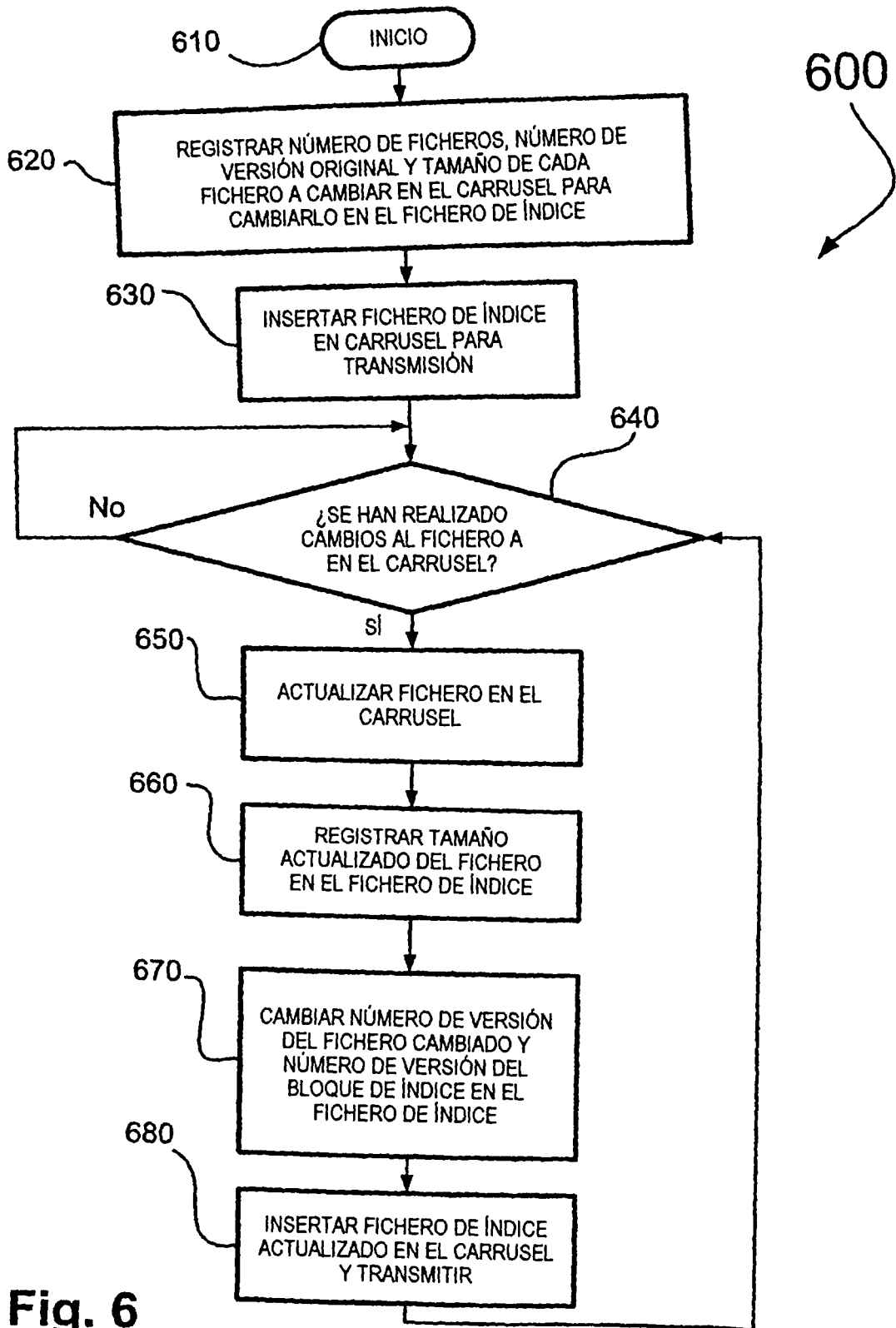


Fig. 6

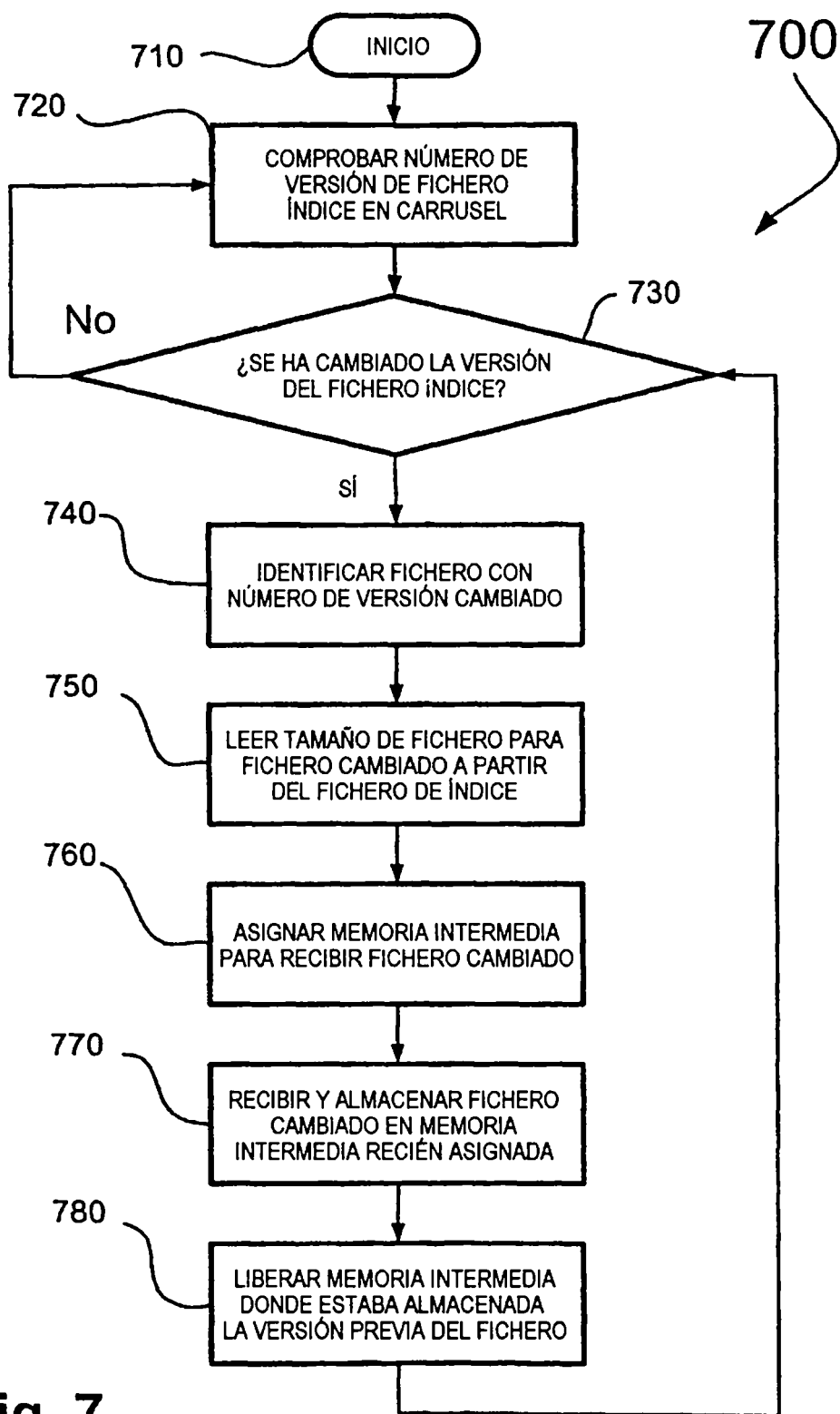


Fig. 7

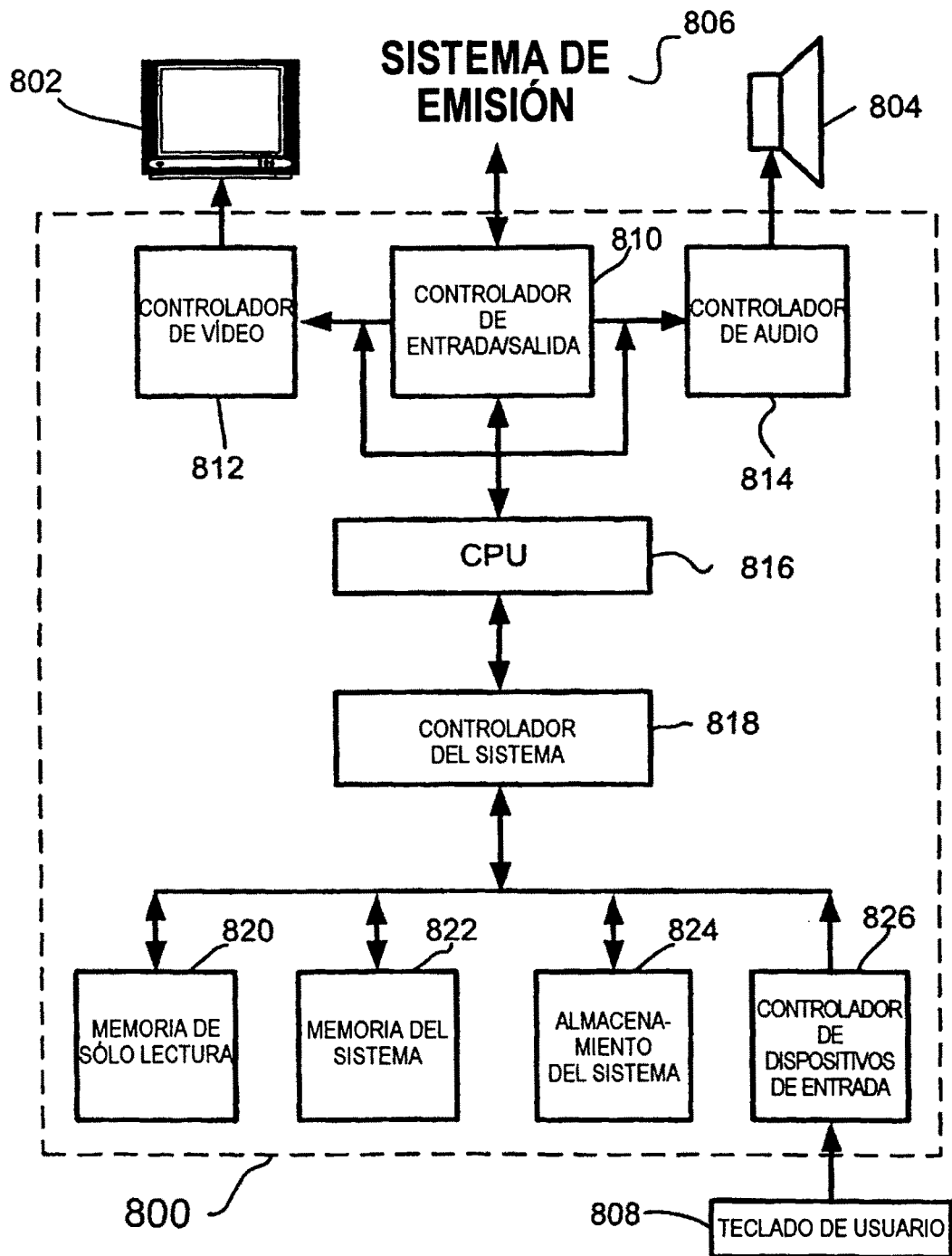


Fig. 8

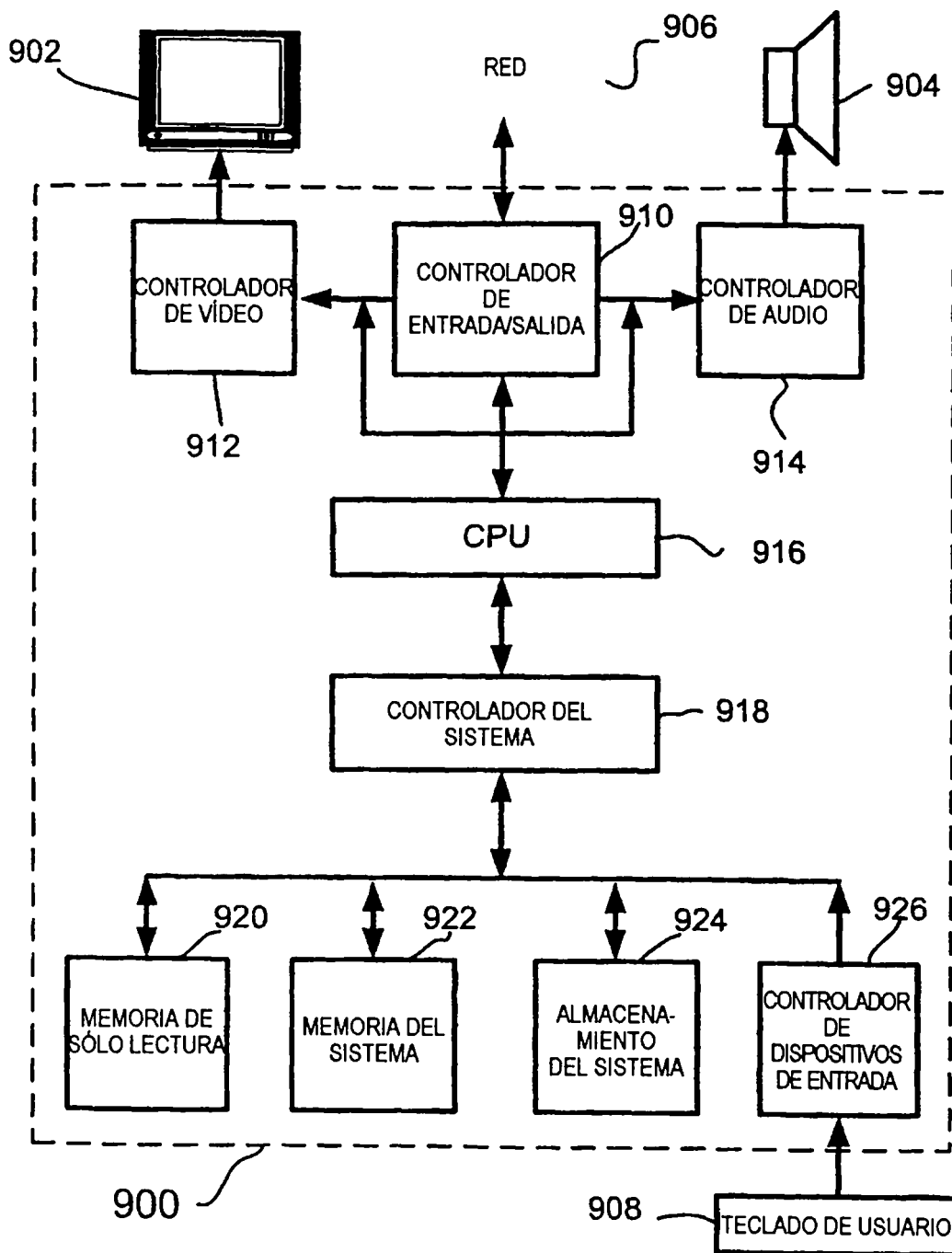


Fig. 9