

I.E.



Industria y Comercio
SUPERINTENDENCIA

SOLICITUD
PATENTE DE INVENCION

21. EXPEDIENTE Nº

04-16521

54. TÍTULO

UN METODO PARA DEMORAR EL ASEGURAMIENTO DE ARCHIVOS DE

SERVIDOR EN EDICION

51. CLASIFICACIÓN INTERNACIONAL

G06F 17/60

71. SOLICITANTE

MICROSOFT CORPORATION

DOMICILIO

REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA

74. APODERADO

DARIO CARDENAS NAVAS

22. BOGOTÁ, D. C.

2004-11-25

2020 3P

PETITORIO

Industria y Comercio
SUPERINTENDENCIA



04 016621 00

SUPERINDUSTRIA Y COMERCIO
Radicación : 04016621 00000000 Folios: 71
Fecha (MM): 2004-02-25 14:33:14
Trámite : 002 PATENTES D 1 REGISTRO/ 411 PRESENTAR
Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

FORMULARIO UNICO DE SOLICITUD DE PATENTE 2000-3

1 SOLICITUD DE: ☒ Patente de Invención ☐ Patente de Modelo de Utilidad			
2 SOLICITANTE (74)	Nombre: MICROSOFT CORPORATION Dirección: One Microsoft Way Redmond, Washington 98052, Estados Unidos de América Teléfono: 3123600 Fax: 3122410 E-Mail: general@cardenasycardenas.com País: REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA	IDENTIFICACION C.C. ☐ NIT ☐ C.E. ☐ Otro ☒ Número:	
3 REPRESENTANTE O/APODERADO (74)	Nombre: DARIO CARDENAS NAVAS Dirección: Carrera 7 No 71-52 Torre B Piso 9 Teléfono: 3123600 Fax: 3122410 E-Mail: general@cardenasycardenas.com	IDENTIFICACION C.C. ☒ NIT ☐ C.E. ☐ Otro ☐ Número: 17.086.829 BTA TP 1.250 C.S.J.	
4 INVENTOR(ES) (72)	Nombre: 1. Mark S. Eichen 2. Mattie 3. Hans E. Heller Dirección: 1. 350 North 28th Street Seattle, Washington 98109 2. 4200 15th St. Avenue Apartment # 280 Redmond, Washington 98052 3. 304 12th Avenue East # 202 Seattle, Washington 98107 Teléfono: 3123600 Fax: 3123600 E-Mail: general@cardenasycardenas.com País: 1. WASHINGTON, ESTADOS UNIDOS DE AMERICA 2. WASHINGTON, ESTADOS UNIDOS DE AMERICA 3. WASHINGTON, ESTADOS UNIDOS DE AMERICA	IDENTIFICACION C.C. ☐ NIT ☐ C.E. ☐ Otro ☒ Número:	
5 Título(54): UN METODO PARA DEMORAR EL ASEGURAMIENTO DE ARCHIVOS DE SERVIDOR EN EDICION			
6 Prioridad SI ☒ NO ☐	(33) País de Origen ESTADOS UNIDOS DE AMERICA	(32) Fecha Febrero 28, 2003	(31) N° de Solicitud 10/376,716
7 Para publicar a partir de la fecha de la presente solicitud a los: 6 meses ☐ 12 meses ☐ 18 meses ☐ Otro ☐			
8 Compromiso de Pago No: 0410300704-10241		Fecha: Febrero 10, 2004	

25-08-05

9

ANEXOS

- ☒ Comprobante de pago de la tasa de presentación de la solicitud.
- ☐ Comprobante de pago de la tasa por concepto de excedente de palabras en la publicación.
- ☒ Comprobante de pago de la tasa por concepto de reivindicación de prioridad. (No04-10.511)
- ☒ Poderes, si fuere el caso. (Protocolo No 16.069)
- ☒ Certificado de existencia y representación legal cuando el solicitante sea persona jurídica. Protocolo No 16.069)
- ☐ Copia certificada de la primera solicitud, si se reivindica prioridad.
- ☒ Traducción simple de la primera solicitud, si se reivindica prioridad.
- ☒ Documento de cesión del inventor al solicitante o a su causante.
- ☒ Descripción de la invención.
- ☒ Una o más reivindicaciones.
- ☒ Dibujos y/o planos necesarios.
- ☐ De ser el caso, copia del contrato de acceso.
- ☐ De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de la comunidad de las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico.
- ☒ Arte final 12x12 y 6x6 cm.
- ☒ Resumen y extracto para publicación.

10

NOMBRE:

DARIO CARDENAS NAVAS



FIRMA:

C.C. 17.066.629 BTA

TP. 1.250 
C.SJ.

SUPERINTENDENCIA Y COMERCIO
Radiacion : 0416521 0000000 Folios /1
Fecha (MM/YY) 2004-02-25 14:38:14
Tramite : 002 PATENTES 0 1 REGISTRO/ 411 PRESENTAC
Dependencia: 200 DIVISION DE NUEVAS CREACIONES

NIT : 000176089-2

RECIBO OFICIAL DE CAJA : 04 - 10,506
FECHA : FEBRERO 10 DE 2004

***** CONSIGNACION *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I. CONSIGNACION		BANCO POPULAR	050-00110-6	21079195	10/02/2004	3,451,000.00

***** CONCEPTO *****

CANT. HISTORICO	CONCEPTO	TOTAL CONCEPTO
-----------------	----------	----------------

1 50005-01-01 SOLICITUDES

1 TRAMITES DE SOL. DE PATENTE DE IN

422,000.00

TOTAL :

422,000.00

SON: CUATROCIENTOS VEINTIDOS MIL PESOS

Un método para demorar el
aseguramiento de archivos
de servir en edición

RESPONSABLE : _____

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

RECIBO A CARDENAS
ABRIL 2004

SUMATORIA Y CANCELACION
Radicacon : 04016321 00000000
Fecha (MM): 2004-02-25 14:30:14
Framite : 002 PAGOS 001 AGENCIA 001 PAGO
Dependencia: 0000 DIVISION DE AGENCIAS CANCELACIONES

NIT : 000176089-2

RECIBO OFICIAL DE CAJA : 04 - 10,311
FECHA : FEBRERO 10 DE 2004

***** CONSIGNACION *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PAGO	Vr. PAGO
CARDENAS Y CARDENAS P.I.	CONSIGNACION	BANCO POPULAR	050-00110-6	21079195	10/02/2004	3,451,000.00

***** CONCEPTO *****

CANT. RENTISTICO	CONCEPTO	TOTAL CONCEPTO
------------------	----------	----------------

1 50005-01-01 SOLICITUDES

82 INVOCACION DE UNA PRIORIDAD EN NU

118,000.00

TOTAL :

118,000.

SON: CIENTO DIEZ Y OCHO MIL PESOS

Un método para demostrar el
aseguramiento de archivos
de servidor en educación

RESPONSABLE : _____

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

CARDENAS
CARDENAS

**UN MÉTODO PARA DEMORAR EL ASEGURAMIENTO DE ARCHIVOS DE
SERVIDOR EN EDICIÓN**

CAMPO DE LA INVENCION

Esta invención se relaciona con el campo de manejo de documentos, y en particular con un método y sistema para demorar el aseguramiento de archivos de servidor que están siendo accesados por un usuario.

ANTECEDENTES DE LA INVENCION

Los computadores servidores tales como los servidores de red pueden desarrollar una doble tarea. Un servidor se puede utilizar como un sitio de publicación en donde un usuario puede acceder contenido. Un servidor también se puede utilizar como un ambiente de colaboración, donde dos o más usuarios pueden colaborar en un documento. Cada usuario puede actualizar o editar de cualquier otra manera el documento. Cuando se utiliza un servidor como un sitio de publicación, este puede suministrar una copia individual de un archivo a cualquier usuario que solicite el archivo, manteniendo la versión en el servidor sin



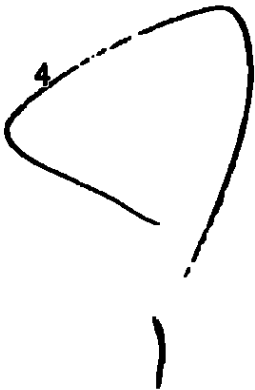
cambios. Cuando se utiliza un servidor como un ambiente de colaboración, este puede permitirle a un usuario que desee editar el archivo acceder a la copia más reciente. A menudo, sin embargo, se utiliza un servidor tanto para estos propósitos de una vez soportando tanto la publicación como las funciones de colaboración.

Tal uso doble crea un problema. Por ejemplo, si el servidor es un servidor de red, el mecanismo por defecto para acceder un archivo en el servidor de red (al seguir un hipervínculo) ejecutará solo una de dos acciones. Dependiendo del hipervínculo y el servidor de red bien sea obturando sobre el hipervínculo se creará una copia del archivo que el usuario puede leer sin alterar el archivo original u obturando sobre el hipervínculo se abrirá el archivo en el servidor para permitirle al usuario editar el archivo. Cuando el archivo está abierto para edición este se asegura, y ningún otro usuario le es permitido editar el archivo.

De manera general, un archivo se puede abrir con varios permisos otorgados al usuario que abre el archivo. Cuando se abre el archivo para lectura el archivo se abre solo lectura, lo cual le permite al usuario leer/barrer otra

forma ver o usar los datos en el archivo, en tanto que los datos no se cambien. Cuando se abre para edición, el archivo se abre lectura/escritura, lo que le permite al usuario los mismos privilegios, pero también le permite al usuario modificar los datos o abrirllos. Cuando el archivo se abre lectura/escritura, el archivo se asegura. Este seguro asegura que solo un usuario está modificando el archivo a la vez, evitando modificaciones de conflicto por dos o más usuarios.

El problema surge cuando la acción por defecto no es lo que el usuario desea, por ejemplo, si el usuario desea editar el archivo, si el defecto está abriendo una copia para lectura, la funcionalidad deseada para el usuario no se suministra. Similarmente, si el defecto está abriendo el archivo para edición, pero el usuario solo desea leer el archivo, el archivo se asegura y otro usuario que desee editar el archivo le será impedido editar el archivo. Un usuario que abre el archivo para lectura y luego desea editar el archivo debe salir del archivo y luego reabrirlo para edición.



8

RESUMEN DE LA INVENCION

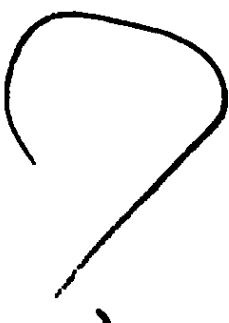
Se implementa un servidor con una acción de apertura de archivo modificado, el cual cuando el usuario desarrolla la apertura modificada, inicialmente abre el archivo sin asegurar el archivo. Cuando el usuario indica (bien sea explícita o implícitamente) que el usuario está tratando o intentando abrir el archivo, el archivo se puede asegurar para edición. De esta manera, la acción por defecto cuando un usuario solicita un archivo es abrir el archivo sin denegarle a otros usuarios el acceso al archivo. Luego, cuando el usuario indica que debe ocurrir la edición, se obtiene el seguro para el archivo.

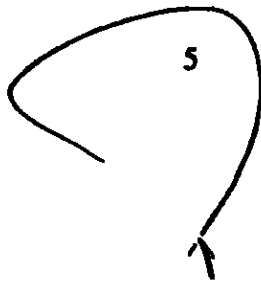


BREVE DESCRIPCION DE LOS DIBUJOS

1

El resumen anterior, así como también la siguiente descripción detallada de las modalidades preferidas, se entienden mejor cuando se leen en conjunto con los dibujos finales. Para el propósito de ilustrar la invención, se muestra en los dibujos modalidades de ejemplo de la invención; sin embargo, la invención no





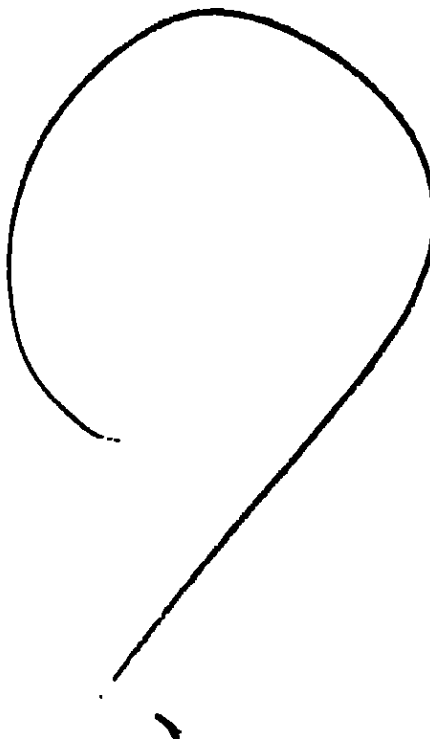
está limitada a los métodos específicos y a las instrumentalidades descritas. En los dibujos:

La Figura 1 es un diagrama de bloques que muestra un ambiente computacional de ejemplo en el cual se pueden implementar los aspectos de la invención;

La Figura 2 es un diagrama funcional que ilustra un ejemplo con una relación entre un ordenador del servidor, una red, un ordenador de usuario y otros ordenadores;

La Figura 3 es un diagrama de flujo que muestra el aseguramiento retrazado de acuerdo con una modalidad de la presente invención; y

La Figura 4 es un diagrama de flujo que muestra el aseguramiento retrazado de acuerdo con una modalidad de la invención.



DESCRIPCIÓN DETALLADA DE LAS MODALIDADES DE ILUSTRACIÓN

Repaso

Con el fin de permitir flexibilidad cuando un usuario abre un documento sobre un servidor compartido, se implementa una apertura modificada. Cuando la apertura modificada se utiliza para abrir un archivo el archivo se abre para solo lectura. Solo cuando y si el usuario que abrió el archivo utilizando la apertura modificada indica que se debe hacer una edición del archivo se obtiene el seguro en el archivo, haciendo el archivo lectura/escritura para el usuario, y asegurando que otros usuarios puedan obtener privilegios de edición para ese archivo.

Dispositivo computacional de ejemplo

La Figura 1 y la siguiente discusión pretenden suministrar una breve descripción general de un ambiente de computación adecuado en el cual la invención se puede desarrollar. Se debe entender, sin embargo, que las agendas, los portátiles y otros dispositivos de computación y objetos de computación de todas las clases

se contemplan para el uso en relación con la presente invención, como se describió anteriormente. De esta manera, aunque un ordenador de propósito general se describe adelante, este es solo un ejemplo, y la presente invención se puede desarrollar con otros dispositivos de cómputo, tal como un cliente fino que tiene interoperabilidad e interacción red/bus. De esta manera la presente invención se puede implementar en un ambiente de servicios de huésped en red en los cuales muy pocos o mínimos recursos de cliente están implicados, por ejemplo, un ambiente de red en el cual el dispositivo de cliente sirve simplemente como una interface a la red/bus tal como un objeto ubicado en un aparato, u otro dispositivo de computo y los objetos también. En esencia en cualquier parte en que se pueden almacenar los datos o de los cuales se puedan retirar los datos es un ambiente deseable o adecuado para operación de acuerdo con la invención.

Aunque no se requiere la invención se puede implementar por vía de un sistema de operación, para uso por un desarrollador de servicios para un dispositivo u objeto, y/o incluido dentro de uso soporte lógico de aplicación que opera de acuerdo con la invención. El soporte lógico

se puede describir en el contesto general de instrucciones ejecutables en ordenador, tal como módulos de programa, que son ejecutados por uno o más ordenadores, tales como la estaciones de trabajo de cliente, los servidores u otros dispositivos. De manera general los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos y similares que desarrollan tareas particulares o implementan tipos de datos abstractos particulares. Típicamente, la funcionalidad de los módulos de programa se puede combinar o distribuir según se deseen en varias modalidades. Más aun, aquellos expertos en la técnica apreciaran que la invención se puede practicar con otras configuraciones por ordenador. Otros sistemas de computación bien conocidos, ambientes y/o configuraciones que pueden ser adecuados para el uso con la invención incluyen pero no están limitados a, ordenadores personales (PC), maquinas de cajeros automáticos, ordenadores servidores, agendas o dispositivos portátiles de computación, sistemas multiprocesador, sistemas basados en microprocesador, electrónicos de consumidor programables, PC de red, aparatos, luces, elementos de control ambiental, microcomputadores, microordenadores y similares. La invención también se puede practicar en la

distribución de ambientes de computación donde las tareas son desarrolladas por dispositivos de procesamiento remoto que están ligados a través de una red/bus de comunicaciones u otro medio de transmisión de datos. En un ambiente de computación distribuido, los módulos de programa se pueden localizar tanto en medios de almacenamiento de computador local como remoto incluyendo dispositivo de almacenamiento de memoria, y nodos de cliente puede a su vez comportarse como nodos de servidor.

La Figura 1 ilustra así un ejemplo de un ambiente de sistema de computación adecuado si en el cual la invención se puede implementar, aunque como se hizo claridad anteriormente, el ambiente de sistema de computación 100 es solo un ejemplo de un ambiente de computación adecuado y no pretende sugerir ninguna limitación en el alcance del usuario o la funcionalidad de la invención. Tampoco se debe interpretar el ambiente de computación 100 como teniendo alguna dependencia o requerimiento que se relacione con cualquiera o combinación de componentes ilustrados en el ambiente operativo de ejemplo 100.

Con referencia a la Figura 1, un sistema de ejemplo para implementar la invención incluye un dispositivo de cómputo de propósito general en la forma de un sistema de ordenador 110. Los componentes del sistema de ordenador 110 pueden incluir, pero no están limitados a, una unidad de procesamiento 120, una memoria de sistema 130, y un bus de sistema 121 que acopla varios componentes de sistema a la unidad de procesamiento 120. el bus del sistema 121 puede ser cualquiera de los varios tipos de estructura de bus incluyendo un bus de memoria o un controlador de memoria, un bus periférico, y un bus local utilizando cualquier variedad de arquitectura de bus. Por vía de ejemplo, y no de limitación, tales arquitecturas incluyen bus de Arquitectura Estándar de Industria (ISA), bus de Arquitectura de Micro Canal (MCA), bus ISA mejorado (EISA), bus local de la Asociación de Estándares de Electrónicos de Videos (VESA), y bus de Interconexión de Componente Periférico (PCI) (también conocido como bus mezanine).

El sistema de ordenador 110 típicamente incluye una variedad de medios legibles en computador. Los medios legibles en computador pueden ser cualquier medio disponible que se pueda acceder mediante el sistema de

computo 110 e incluye tanto medio volátiles como no volátiles, medios removibles como no removibles. Por vía de ejemplo, y no de limitación, un medio legible de computadora puede comprender un medio de almacenamiento de computadora y un medio de comunicación. Los medios de almacenamiento de computadora incluyen medios volátiles y no volátiles, removibles y no removibles implementados en cualquier método o tecnología para almacenamiento de la información tal como las instrucciones legibles por ordenador, las estructuras de datos, los módulos de programa u otros datos. Los medios de almacenamiento por ordenador incluyen, pero no están limitados a, Memoria de Acceso Aleatorio (RAM), memoria de solo lectura (ROM), memoria de solo lectura programable electrónicamente borrrable (EEPROM); memoria de destello u otra tecnología de memoria. La memoria de solo lectura de disco compacto (CD ROM), los discos versátiles digitales (DVD) u otros dispositivos de disco óptico, cartuchos magnéticos, cintas magnéticas, almacenamientos de disco magnético u otros dispositivos de almacenamiento magnético o cualquier otro medio que se pueda utilizar para almacenar la información deseada y que se pueda acceder mediante al sistema de ordenador 110. Los medios de comunicación típicamente tienen modalidades de instrucción medial por

ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora u otro mecanismo de transporte e incluye cualesquier medios de suministro de información. El término "señal de datos modulada" significa una señal que tiene una o más de sus características establecidas o cambiadas de tal manera que codifica la información en la señal. Por vía de ejemplo y no de limitación. Los medios de comunicación incluyen medios alámbricos tales como redes alámbricas o conexión alámbrica directa y medios inalámbricos tales como acústicos, RF, infrarrojos u otros medios inalámbricos. Las combinaciones de cualquiera de las anteriores se deben incluir dentro del alcance de los medios legibles por ordenador.

La memoria del sistema 130 incluye un medio de almacenamiento por ordenador en la forma de memoria volátil y/o no volátil tal como memoria de solo lectura (ROM) 131 y memoria de acceso aleatorio (RAM) 132. Un sistema básico de entrada/salida 133 (BIOS), que contiene las rutinas básicas que ayudan a transferir la información entre elementos dentro del sistema del ordenador 110, tal como durante el arranque, se almacena típicamente en la ROM 131. La RAM 132 típicamente

contiene los módulos de datos y/o programa que son inmediatamente accesibles y/o están siendo en el momento operados por una unidad de procesamiento 120. Por vía de ejemplo y no de limitación, la Figura 1 ilustra el sistema operativo 134, los programas de aplicación 135, u otros módulos de programa 136, y los datos de programa 137.

El sistema de ordenador 110 también puede incluir otros medios de almacenamiento de ordenador removibles/no removibles, volátiles/no volátiles. Por vía de ejemplo solamente, la Figura ilustra una unidad de disco duro 141 que lee o escribe a medios magnéticos no removibles, no volátiles, una unidad de disco magnético 151 que lee o escribe a disco magnético removible no volátil 152, y una unidad de disco óptico 151 que lee o escribe a disco óptico removible, no volátil 156, tal como el CD ROM u otro medio óptico. Otros medios de almacenamiento de ordenador removibles/no removibles, volátiles/no volátiles que se pueden utilizar en el ambiente operativo de ejemplo incluyen, pero no están limitados a, cartuchos de cinta magnética, tarjetas de memoria de destello, discos versátiles digitales, cintas de video digitales, RAM de estado sólido, ROM de estado sólido y similares.

La unidad de disco duro 141 está típicamente conectada al bus del sistema 121 a través de una interface de memoria no removible tal como una interface 140, una unidad de disco magnético 151 y una unidad de disco óptico 155 que está típicamente conectadas al bus del sistema 121 mediante una interface de memoria removible, tal como la interface 150.

Las unidades y sus medios de almacenamiento de ordenador asociado discutidas anteriormente e ilustradas en la Figura 1 suministran almacenamiento de las instrucciones legibles por ordenador, las estructuras de datos, los módulos de programa y otros datos para el sistema de ordenador 110. En la Figura 1, por ejemplo, la unidad de disco duro 141 se ilustra como un sistema operativo de almacenamiento 144, programas de aplicación 145, u otros módulos de programa 146 y datos de programa 147. Nótese que estos componentes pueden ser iguales o diferentes del sistema operativo 134, los programas de aplicación 135, u otros módulos de programa 136, y los datos de programa 137. El sistema operativo 144, los programas de aplicación 145 u otros módulos de programas 146, y los datos del programa 147 son dados diferentes números aquí para ilustrar que, como mínimo, ellos son copias.

to file
w/program

diferentes. Un usuario puede ingresar los comandos y la información dentro del sistema de ordenador 110 a través de los dispositivos de entrada tal como el teclado 162 y el dispositivo de indicación 161, comúnmente denominado como ratón, la esfera de seguimiento "trackball" o la almohadilla "touch pad". Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, un dispositivo de juegos "joystick", una almohadilla de juegos "game pad" un disco satélite, un explorador, o similar. Estos y otros dispositivos de entrada están a menudo conectados a la unidad de procesamiento 120 a través de una interface de entrada de usuario 160 que está acoplado al que está acoplado al bus del sistema 121, pero se pueden conectar por otras estructuras de interface y bus, tal como el puerto paralelo, el puerto de juegos o el bus serial universal (USB). Una interface grafica 182, tal como la Northbridge, también se puede conectar la bus de sistema 121. La Northbridge es un conjunto de chips que comunican con la CPU, o la unidad de procesamiento huésped 120, y asume la responsabilidad de las comunicaciones del puerto grafico acelerado (AGP). Una o más unidades de procesamiento grafico (GPU) 184 pueden comunicar con la interface grafica 182. A este respecto, las GPU 184 generalmente incluyen

almacenamiento de memoria en chip, tal como almacenamiento de registro y las GPU 184 comunican con una memoria de video 186. Las GPU 184, sin embargo, son solo un ejemplo de un coprocesador y así una variedad de dispositivos de coprocesamiento se pueden incluir en el sistema de ordenador 110. Un monitor 191 u otro tipo de dispositivo de exhibición también está conectado al bus del sistema 121 por vía de una interface, tal como una interface de video 190, que puede a su vez comunicar con la memoria de video 186. Además del monitor 191, el sistema de ordenador también puede incluir otros dispositivos de salida periféricos tales como parlantes 197 e impresora 196, que pueden ser conectados a través de la interface periférica de salida 195.

El sistema de ordenador 110 puede operar en una red o en un ambiente distribuido utilizando conexiones lógicas a uno o más ordenadores remotos, tal como el ordenador remoto 180. El ordenador remoto 180 puede ser un ordenador personal, un servidor, un enrutador, un PC de red, un dispositivo homologo u otro nodo de red común, y típicamente incluye muchos o todos de los elementos descritos anteriormente con relación al sistema de ordenador 110, aunque solo un dispositivo de

almacenamiento de memoria 181 se ha ilustrado en la Figura 1. Las conexiones lógicas descritas en la Figura 1 incluyen una red de área local (LAN) 171 y una red de área amplia (WAN) 173, pero muchos también incluyen otras redes/buses. Tales ambientes de red son comunes en los hogares, oficinas, redes amplias de ordenadores de empresa, Intranet e Internet.

Cuando se utiliza en un ambiente de red LAN, el sistema de ordenador 110 está conectado al LAN 171 a través de una interface o adaptador de red 170. Cuando se utiliza en un ambiente de red WAN el sistema de ordenador 110 típicamente incluye un módem 172 u otros medios para establecer comunicación sobre el WAN 173, tal como el Internet. El módem 172, que puede ser interno o externo, puede estar conectado al bus del sistema 121 por vía del interface entrada de usuario 160, otro mecanismo apropiado. En el ambiente de red, los módulos de programa descritos con relación al sistema de ordenador 110, o porciones de estos, se pueden almacenar en el dispositivo de almacenamiento de memoria remoto. Por vía de ejemplo, y no delimitación, la Figura 1 ilustra programas de aplicación remoto 185 como residentes en el dispositivo de memoria 181. Se apreciará que las condiciones de red

mostradas son ejemplos y se puede utilizar otros medios para establecer un vínculo de comunicaciones entre los ordenadores.

Varios entramados de computación distribuidos han sido desarrollados a la luz de la convergencia de la computación personal y el Internet. Los usuarios individuales y de negocios son suministrados de manera similar con interfaces interoperables y posibilitados por red para aplicaciones y dispositivos de cómputo, haciendo las actividades de computación crecientemente orientadas a navegador de red o de cadena.

Por ejemplo, la plataforma MICROSOFT®'s NET incluye servidores, servicios de bloques de construcción tales como el soporte lógico de dispositivos de almacenamiento de datos basados en red y bajables. Mientras que las modalidades de ejemplo aquí son descritas en relación con el soporte lógico que reciben un dispositivo de computo, una o más porciones de la invención también se puede implementar por vía de sistema operativo, la interface de programación de aplicación (API) o un objeto para un "hombre medio" entre cualquiera de un coprocesador, un dispositivo de exhibición y un objeto que se solicita,

como tal operación de acuerdo con la invención se puede desarrollar por, soportar en o acceder por vía de todos los lenguajes de red y los servicios, y en otros entramados de computación distribuida también.

Seguro de Demora

Con referencia a la Figura 2, un archivo 210 reside en un ordenador servidor 200. Un usuario con permiso para hacer una solicitud para abrir el archivo. El usuario puede estar conectado al ordenador servidor de numerosas maneras. Por ejemplo, el usuario puede conectar directamente, vía una red de área local (LAN) o remotamente por vía de una red tal como el Internet. En la Figura 2 de ejemplo, el ordenador del usuario 220 está conectado por vía de una red 230 al ordenador servidor 200.

El usuario solicita que el archivo se abra. Un método posible para presentar tal solicitud es al seleccionar un hipervínculo sobre la página de red servida por el ordenador servidor 200.

Cuando la solicitud se recibe por el servidor, el servidor le permite al computador del usuario 220 abrir el archivo. El archivo se abre en el ordenador del usuario mediante una aplicación 240. Por ejemplo, esta aplicación puede ser una aplicación de procesador de palabra, una aplicación de hoja de cálculo, una aplicación de reproductor de audio, o cualquier aplicación mediante la cual el usuario pueda utilizar los datos que están siendo descargados. En una modalidad, el usuario selecciona un hipervínculo en una página de red que se está viendo en una aplicación de navegación de red en el ordenador del usuario 220, y la aplicación de navegación de red abre el archivo con la aplicación apropiada.

Sec 19
procto

Cuando el archivo se abre, en una modalidad, se hace una copia del archivo en el caché del ordenador del usuario 230. es la copia en el ordenado del servidor 220 lo que el usuario navega o lee (o con el cual hace interacción- por ejemplo, al reproducirlo en un reproductor de audio si el archivo incluye datos de audio). Cuando el archivo es abierto inicialmente, el archivo se abre con permisos de "solo lectura". De esta manera más de un usuario puede acceder el archivo y usarlo.

En una modalidad alternativa, en lugar de que se haga una copia local en el ordenador del usuario 230 todos los usuarios con permiso de solo lectura están viendo el archivo en el servidor.

El aseguramiento del archivo solo ocurre luego de una indicación de que el usuario pretende evitar el archivo. El seguro le permitiría al usuario tener permisos de "lectura/escritura" en el archivo. El usuario podría entonces modificar o anexar datos de archivo.

Este intento de editar indicación puede ser uno de un número de diferentes cosas. En una modalidad, el intento de editar se indica por el usuario modificando el archivo de alguna manera. En otra modalidad, el intento de editar se indica por el usuario. Esto puede ocurrir a través de un comando de "edición" especializado incorporado en la aplicación, o cuando el usuario solicita que se guarden cambios ya realizados. En otra modalidad, el intento de editar la determinación es una determinación de que el usuario ha tenido dicho archivo abierto durante un periodo de tiempo preespecificado. Por ejemplo, si el usuario ha tenido el archivo abierto en modo de solo

lectura durante diez minutos, se puede inferir que el usuario pretende editar el archivo.

En una modalidad, cuando se obtiene el seguro, se hace la determinación de que si la copia del archivo en el caché del ordenador del usuario 230 es la misma versión de la versión habitualmente en el servidor. Esto asegura que no ha ocurrido ninguna edición durante este tiempo. Por ejemplo, el usuario A abre la primer versión del archivo, pero no demuestra un intento de edición (y por lo tanto no dispara el seguro). Si el usuario A dispara entonces el seguro al demostrar un intento de edición, este se debe asegurar de que la versión del archivo habitualmente en el servidor es la misma que la que el usuario A está viendo o pretende editar. Si el usuario B ha abierto y modificado el archivo en el ordenador servidor 200 en algún punto entre la abertura del usuario A del archivo y la demostración del usuario A del intento de edición, puede ocurrir un conflicto. Las opciones de resolución del conflicto se pueden presentar al usuario.

Por ejemplo, el usuario A puede hacer cambios sustanciales al archivo, y puede decidir que los cambios deben suplantarse cualquier cambio que ha ocurrido en el

r

interin. La versión del ordenador de usuario 230 es luego escrita al ordenador servidor 200 como archivo 210, y el archivo se asegura de tal forma que el usuario A tenga privilegios lectura/escritura. Alternativamente el usuario A puede decidir guardar la versión que el usuario A está trabajando localmente como otro archivo, y reabrir el archivo 210 en el ordenador servidor 200 para edición. Otras opciones de resolución de conflicto también se pueden presentar al usuario.

Con el fin de determinar si la copia del archivo en el ordenador del usuario 230 es la misma versión que la versión del archivo 210 habitualmente en el servidor, la información de la versión se puede almacenar y comparar. Esto puede incluir un número de versiones, tiempo y fecha de información guardada, u otros datos de la versión. Alternativamente, se puede hacer una comparación completa del archivo actual 210 y la versión de la antememoria en el ordenador del usuario 230 para determinar si los contenidos del archivo son los mismos.

En una modalidad alternativa donde el usuario no copia el archivo 210 al ordenador del usuario 230 si no en su lugar lo abre directamente desde el ordenador servidor

r

200, cuando el intento de editar que dispara el evento ocurre, el seguro de archivo de edición 210 es obtenido por el usuario.

Como se muestra en la Figura 3, se recibe una solicitud para abrir el archivo, en la etapa 300. En la etapa 310, se abre el archivo. En una modalidad, el archivo se abre directamente del servidor en modo de solo lectura; en una modalidad alternativa, se hace una copia del archivo y se abre. En la etapa 320, se determina que ha ocurrido un evento de disparo de edición. Este evento de disparo de edición puede ser los cambios cortados por el usuario a una copia local del archivo, cuando se ha hecho una. El evento de disparo de edición también puede ser el intento del usuario de cambiar el archivo, donde la copia del servidor del archivo fue abierta solo para lectura. Este evento de disparo de edición también puede ser que un periodo predeterminado de tiempo ha pasado, una indicación de "edición" recibida del usuario, o cualquier otro evento que necesite el aseguramiento del archivo del servidor para edición.

En la etapa 330, se obtiene el seguro en el archivo. Esto asegura la copia del archivo del servidor para que el

usuario lo modifique si la posibilidad de modificación es con conflicto.

Como se muestra en la Figura 4, la resolución de conflicto se puede implementar en una modalidad. Como en la Figura 3, la primera etapa 410 es cuando se recibe una solicitud para abrir el archivo. En la segunda etapa 420 se hace una copia local del archivo, y esa copia local se abre para edición. En la tercera etapa 430, se establece la determinación de que ha ocurrido un evento de disparo de edición. Se hace una comparación 440 para determinar si la versión del archivo en el servidor es la misma que la versión que se ha copiado en la etapa 420. Si es la misma versión, entonces se obtiene el seguro en archivo, etapa 450. Si el usuario ha hecho cambios a la copia local del archivo, la copia del servidor del archivo se pueda actualizar para reflejar aquellos cambios. Si la versión ha cambiado, opciones de resolución de conflictos se presentan al usuario de la etapa 460. Esto puede conducir a que se obtenga el seguro en el archivo, etapa 450, sin embargo, esto también puede conducir a otras opciones de resolución de conflicto.

De esta manera, el archivo se puede abrir por numerosos usuarios simultáneamente, quienes pueden utilizar todo el contenido del archivo. Sin embargo, cuando un usuario desea modificar el archivo, el usuario obtiene el seguro, y el usuario tendrá privilegios de lectura/escritura para el archivo.

Conclusión

Como se menciona anteriormente, aunque las modalidades de ejemplo de la presente invención se han descrito con relación a varios dispositivos de computo y arquitecturas de red, los conceptos subyacentes se pueden aplicar a cualquier dispositivo o sistema de computo en el cual se deseable implementar una apertura de archivo modificado. De esta manera, los métodos y sistemas de la presente invención se pueden aplicar a una variedad de aplicaciones y dispositivos. Aunque los lenguajes de programación de ejemplo, los nombres y los ejemplos son escogidos aquí como representativos de varias opciones, estos lenguajes, nombres y ejemplos no pretenden ser limitantes. Un experto en la técnica apreciará que hay numerosas formas de suministrar códigos de objeto que

logren sistemas iguales, similares o equivalentes y métodos logrados por la invención.

Las varias técnicas descritas aquí se pueden implementar en relación con el equipo o el soporte lógico o, donde sea adecuado, con una combinación de ambos. Así, los métodos y aparatos de la presente invención, o ciertos aspectos o porciones de estos, pueden tomar la forma de código de programa (es decir instrucciones) con modalidades y medios tangibles, tales como disquetes, CD-ROM, unidades duras, o cualquier otro almacenamiento para lectura de maquina, en donde, cuando el código de programas se carga y ejecuta por la maquina, tal como un ordenador, la maquina se vuelve un aparato para practicar la invención. En el caso de la ejecución de código de programa en ordenadores programables el dispositivo de computo generalmente incluirá un procesador, un medio de almacenamiento libre por el procesador (incluyendo memoria volátil y no volátil y/o elementos de almacenamiento), al menos un dispositivo de entrada, y al menos un dispositivo de salida. Uno o más programas que puedan utilizar los servicios de procesamiento de señal de la presente invención, por ejemplo, a través del uso de un API de procesamiento de datos o similar, son

preferiblemente implementados en un procedimiento de alto nivel o lenguaje de programación orientada a objeto para comunicarse común ordenador. Sin embargo, los programas se pueden implementar en lenguaje de ensamble o de maquina si se desea. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado, y combinado con implementaciones de equipo.

Los métodos y aparatos de la presente invención también se pueden practicar por vía de comunicaciones con modalidades en la forma de código de programa que es transmitido sobre algún medio de transmisión, tal como sobre alambre o cable eléctrico, a través de fibras ópticas, o por vía de cualquier otra forma de transmisión, en donde, cuando el código de programa se recibe y se carga y se ejecuta por una maquina, tal como una EPROM, un arreglo de puerta, un dispositivo lógico programable (PLD), un ordenador de cliente, un grabador de video similar, o una maquina receptora que tiene capacidades de procesamiento de señal como se describió en las modalidades de ejemplo se vuelve un aparato para practicar la invención. Cuando se implementa en un procesador de propósito general, el código de programa se combina con el procesador para suministrar un aparato

único que opera para invocar la funcionalidad de la presente invención. Adicionalmente, cualquier técnica de almacenamiento utilizada en relación con la presente invención puede invariablemente ser una combinación de equipo y soporte lógico.

Aunque la presente invención se ha descrito en relación con las modalidades preferidas de las varias Figuras, se debe entender que otras modalidades similares se pueden utilizar o se pueden hacer modificaciones y adiciones a la modalidad descrita para desarrollar la misma función de la presente invención sin desviarse de esta. Adicionalmente, se debe enfatizar que una variedad de plataformas de ordenador incluyendo sistemas operativos para dispositivos de agenda y otros sistemas de operación específicos de aplicación se contemplan especialmente como el número de dispositivos de redes inalámbricas continua proliferando. Por lo tanto, la presente invención no se debe limitar a ninguna modalidad simple, si no se debe considerar en el ámbito y alcance de acuerdo con las reivindicaciones finales.

REIVINDICACIONES

1. Un método para demorar el aseguramiento del archivo que está siendo abierto, que comprende: *\$ Conrad X.*

Recibir una solicitud para abrir dicho archivo;

Abrir dicho archivo para lectura;

Determinar que un evento de disparo de edición ha ocurrido; y

Obtener un seguro sobre dicho archivo.

2. El método de la reivindicación 1, ²donde dicha apertura del archivo comprende abrir el archivo en modo de solo lectura, y donde dicha obtención de seguro comprende cambiar el modo escritura/lectura para dicho usuario. *u?*

3. El método de la reivindicación 1, donde dicha determinación de que el evento de disparo de

edición ha ocurrido comprende recibir un cambio a dicho archivo.

4. El método de la reivindicación 1, donde dicha determinación que un evento de disparo de edición ha ocurrido comprende recibir una solicitud para editar dicho archivo de un usuario.

5. El método de la reivindicación 1, donde dicha determinación de que un evento de disparo de edición ha ocurrido comprende una determinación de que un usuario ha tenido dicho archivo abierto durante un periodo preespecificado de tiempo.

6. El método de la reivindicación 1, donde dicha etapa de abrir dicho archivo comprende crear una copia de usuario de dicho archivo.

7. El método de la reivindicación 6, donde dicho archivo se localiza en un ordenador servidor, un usuario está utilizando un ordenador de usuario, dicho ordenador de usuario y dicho ordenador de usuario está operablemente conectado, y dicha copia

de usuario de dicho archivo es local a dicho ordenador de usuario.

8. El método de la reivindicación 6, donde dicha etapa de obtener un seguro en dicho archivo comprende:

Determinar si el archivo ha cambiado desde que la copia de usuario se hizo; y si dicho archivo no ha cambiado, obtener un seguro sobre dicho archivo.

9. El método de la reivindicación 8, en donde dicha etapa de determinar si dicho archivo ha cambiado comprende comparar la información de una versión habitual de dicho archivo sobre dicho ordenador de servidor con la información de la versión de dicha copia de usuario.

10. El método de la reivindicación 9, donde dicha información de versión para una versión específica de dicho archivo comprende información relacionada con el tiempo del último cambio a dicha versión específica.

11. El método de la reivindicación 8, donde dicha etapa de obtener un seguro en dicho archivo comprende además:

Si dicho archivo ha cambiado, ofrecer una o más opciones de resolución de conflicto a dicho usuario.

12. El método de la reivindicación 8, donde dichas opciones de resolución de conflicto comprende la opción de guardar la copia de usuario de dicho archivo.

13. El método de la reivindicación 8, donde dichas opciones de resolución de conflicto comprende la opción de retornar dicha versión habitual de dicho archivo al ordenador servidor.

14. Al menos uno de un sistema operativo, un medio legible por ordenador habiendo almacenado sobre este una pluralidad de instrucciones ejecutables por ordenador, un dispositivo de coprocesamiento, un dispositivo de computo, y una señal de datos modulada que lleva las instrucciones ejecutables

por ordenador para desarrollar el método de la reivindicación 1.

154
no
se
permite

15. Un sistema de ordenador para solicitar un archivo desde un servidor, dicho servidor operablemente conectado a dicho sistema de ordenador que comprende:

Un módulo de entrada para recibir una solicitud para abrir el archivo en un ordenador servidor;

Un módulo de apertura de modo de solo lectura para abrir dicho archivo en modo de solo lectura, operablemente conectad a dicho módulo de entrada de usuario;

Un módulo de indicación de disparo de edición para determinar si ha ocurrido un disparo de edición;

Un módulo de apertura de modo de lectura-escritura para abrir dicho archivo en modo de lectura-escritura, operablemente conectado a dicho módulo de indicación de intento de edición; y

reducir
terminos
de
proc. de

?

Un módulo de uso de archivo para utilizar dicho
archivo, operablemente conectado a dicho módulo de
apertura de modo de solo lectura y dicho módulo de
apertura de modo de lectura-escritura.

16. El sistema de la reivindicación 15, donde dicha
conexión operable comprende una conexión de red.

17. El sistema de la reivindicación 15, donde dicho
módulo de indicación de disparo de edición
determina si dicho usuario ha solicitado un cambio
a dicho archivo.

18. El sistema de la reivindicación 15, donde dicho
módulo de indicación de disparo de edición
determina si dicho usuario ha solicitado que dicho
archivo quede disponible para edición.

19. El sistema de la reivindicación 15, donde dicho
módulo de indicación de disparo de edición
determina si dicho usuario ha tenido dicho archivo
abierto durante un periodo de tiempo
preespecificado.

20. El sistema de la reivindicación 15, que comprende
además:

Almacenamiento para almacenar una copia de usuario
de dicho archivo.

21. El sistema de la reivindicación 20, que comprende
además:

Un módulo de detección de cambio para detectar i
dicha copia de usuario de dicho archivo es una
copia de una versión habitual de dicho archivo en
dicho servidor, operablemente conectada a dicho
módulo de lectura-escritura.

22. El sistema de la reivindicación 21, donde dicho
módulo de apertura de modo de lectura-escritura
abre dicho archivo para dicho usuario en modo de
lectura-escritura si no se han detectado cambios en
dicha versión habitual de dicho archivo en dichos
servidor por dicho módulo de detección de cambio.

23. El sistema de la reivindicación 21, que comprende
además:

Un módulo de resolución de cambio para presentar al menos una opción de resolución de cambio a dicho usuario luego de la detección de un cambio en la versión habitual de dicho archivo en dicho servidor.

24. El sistema de la reivindicación 23, donde dicha opciones resolución de conflicto comprende la opción de guardar la copia de usuario de dicho archivo.

25. El sistema de la reivindicación 23, donde dichas opciones de resolución de conflictos comprende la opción de devolver dicha versión habitual de dicho archivo a dicho ordenador servidor.

26. El sistema de la reivindicación 20, donde dicho módulo de detección de cambio compara la información de la versión de una versión habitual de dicho archivo en dicho ordenador servidor con la información de la versión de dicha copia de usuario.

27. El sistema de la reivindicación 26, donde dicha información de versión para una versión específica de dicho archivo comprende información con relación al tiempo del último cambio de dicha versión especificada.

28. Un método para utilizar un archivo remoto en un servidor remoto, que comprende:

Luego de la indicación de que un usuario ha solicitado la abertura de dicho archivo remoto, recibir una copia de la versión habitual de dicho archivo remoto proveniente de dicho servidor remoto;

Mostrar dicha copia para dicho usuario utilizando aplicación de soporte lógico adaptada para el uso con archivos del mismo tipo que dicho archivo remoto;

Luego de la indicación de dicho usuario de que dicho usuario pretende dicho archivo remoto, solicitar que dicho archivo en dicho servidor

39

remoto se asegure para escritura por otros sea asegurado para escritura por otros usuarios.

29. El método de la reivindicación 28, donde se solicita dicha indicación de un usuario que abre dicho archivo remoto comprende que dicho usuario obture sobre un hipervínculo.

30. El método de la reivindicación 28, donde dicha indicación de que el usuario pretende editar dicho archivo remoto comprende recibir un cambio a dicha copia en dicha aplicación.

31. El método de la reivindicación 28, donde dicha indicación de que dicho usuario pretende editar dicho archivo remoto comprende recibir una solicitud para editar dicha copia de dicho usuario en dicha aplicación.

32. El método de la reivindicación 28, donde dicha indicación de que dicho usuario pretende editar dicho archivo remoto comprende una determinación que dicho usuario ha tenido dicha copia abierta durante un periodo preespecificado de tiempo.

?

33. El método de la reivindicación 28, donde dicha etapa de solicitar que dicho archivo en dicho servidor remoto se asegure para escritura por otros usuarios comprende:

Determinar si el archivo remoto se ha cambiado desde que dicha copia se hizo; y

Si dicho archivo remoto no ha cambiado, solicitar que dicho archivo remoto en dicho servidor remoto se asegure para escritura por otros usuarios.

34. El método de la reivindicación 33, donde dicha etapa a determinar si dicho archivo ha sido cambiado comprende comparar la información de la versión de dicho archivo remoto en dicho servidor remoto con la información de la versión de dicha copia.

35. El método de la reivindicación 33, donde dicha etapa de obtener un seguro para dicho usuario comprende además:

Si dicho archivo remoto ha sido cambiado, ofrecer una o más opciones de resolución de conflicto a dicho usuario.

36. Al menos uno de un sistema operativo, un medio libre por ordenador habiendo almacenado sobre este una pluralidad de instrucciones ejecutables por ordenador, un dispositivo de coprocesamiento, un dispositivo de computo y unas instrucciones ejecutables por ordenador que lleva una señal de datos modulada para desarrollar el método de la reivindicación 28.

RESUMEN

Se implementa un servidor con una acción de apertura de archivo modificada, la cual, el usuario desarrolla una apertura modificada, inicialmente abre un archivo sin asegurar el archivo. Cuando un usuario indica (sea explícita o implícitamente) que el usuario está tratando abrir el archivo se puede asegurar para edición. De esta manera, la acción por defecto cuando el usuario solicita un archivo es abrir el archivo sin denegarles a otros usuarios acceso al archivo. Luego, cuando el usuario indica que ha ocurrido la edición, se obtiene el seguro para el archivo.

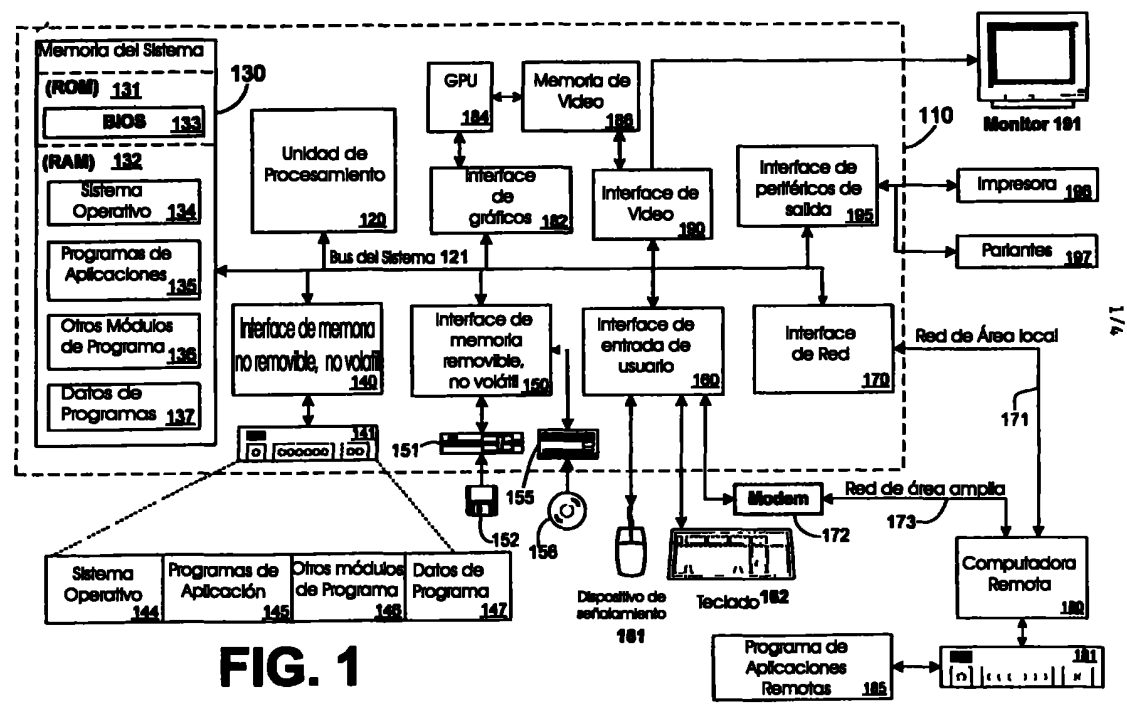


FIG. 1

2/4

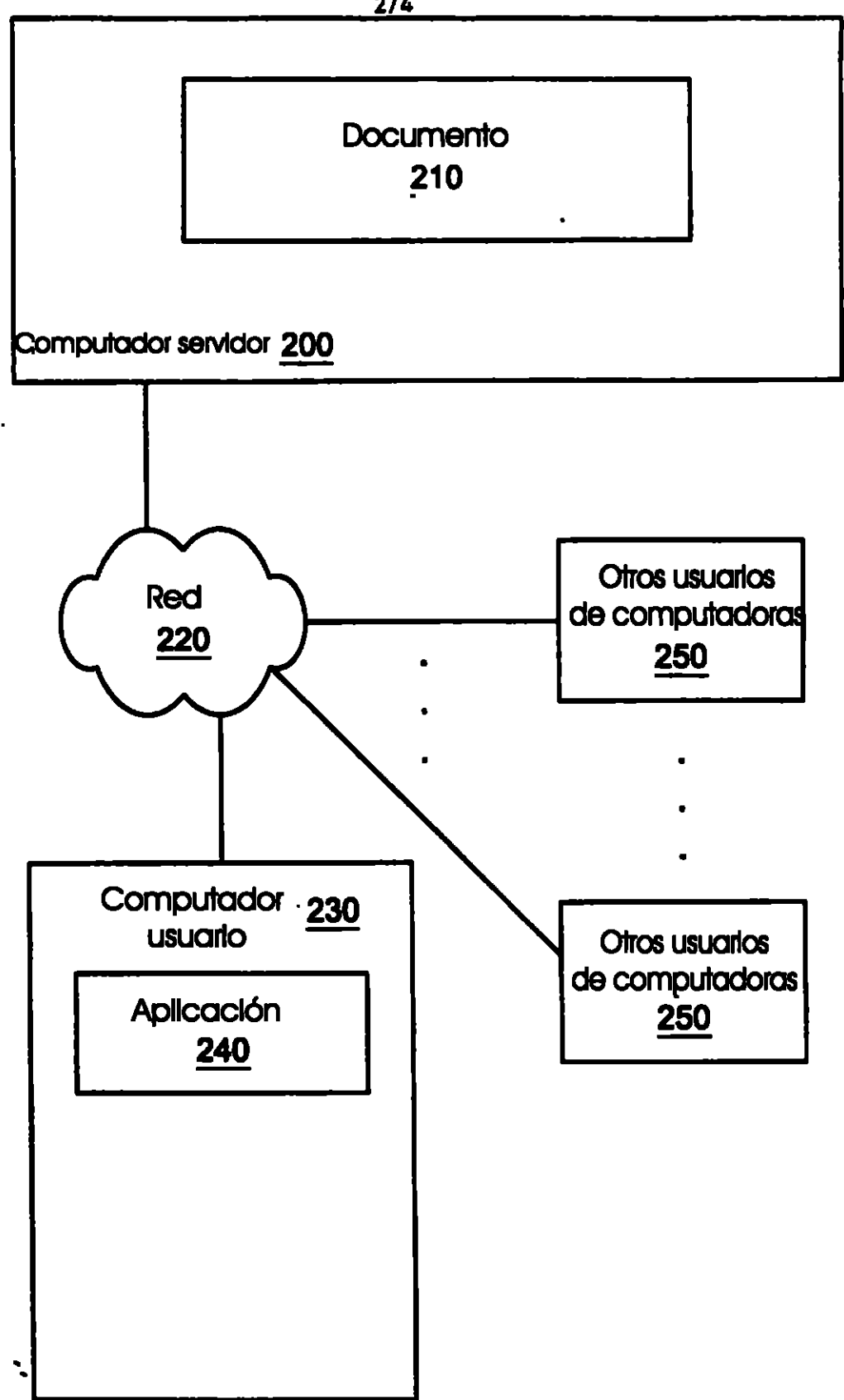


Fig. 2

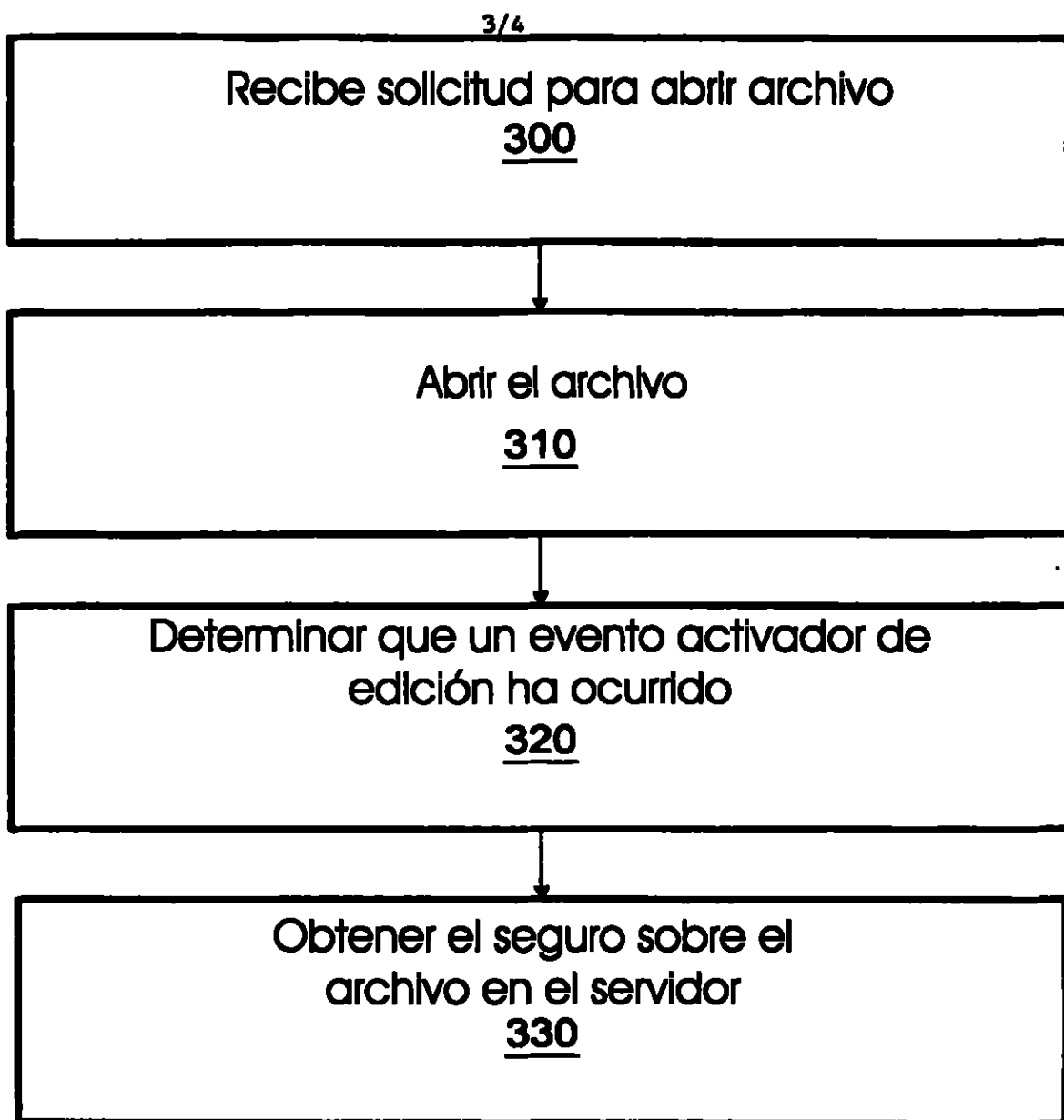


FIG. 3

5020

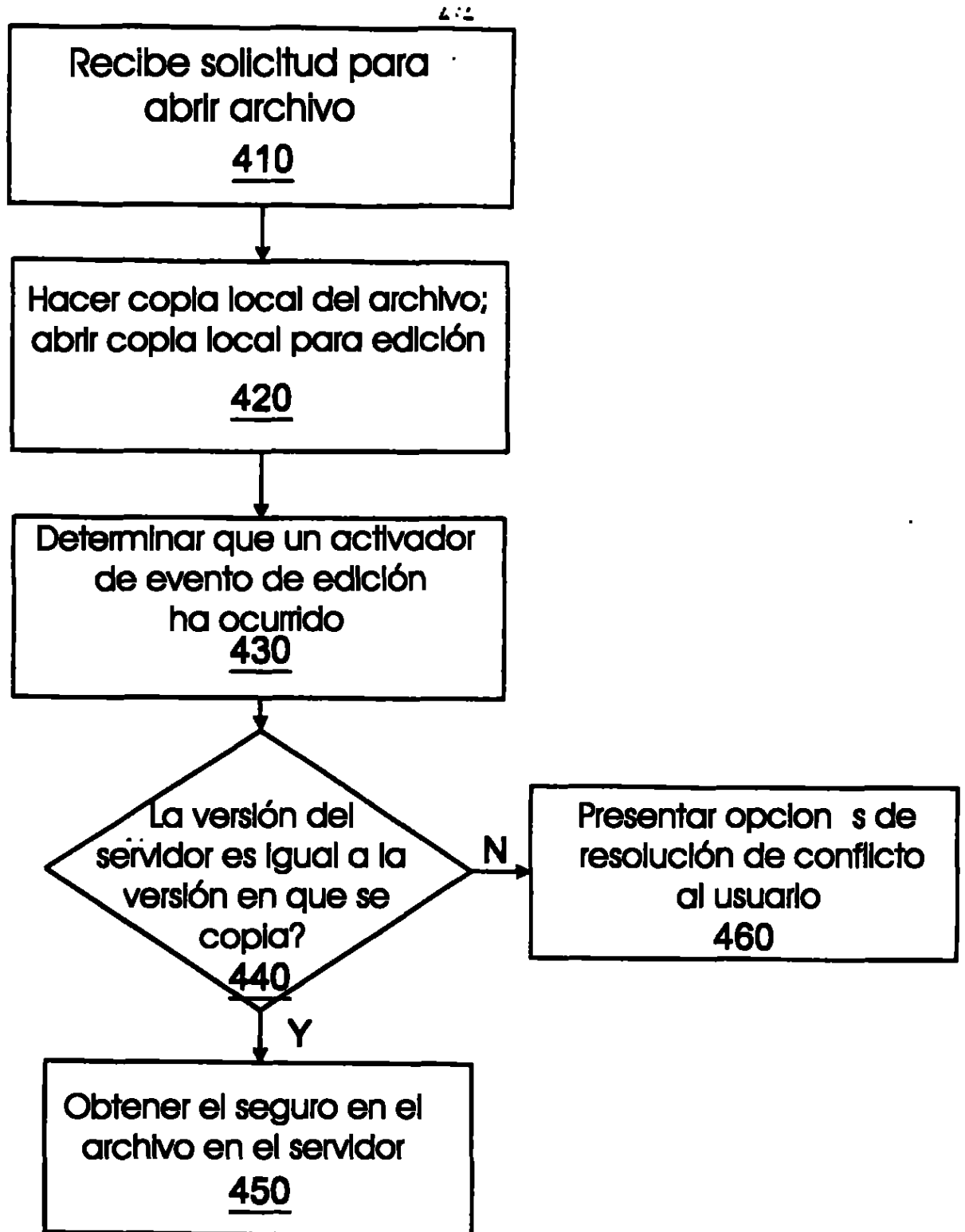


FIG. 4

A METHOD TO DELAY LOCKING OF SERVER FILES ON EDIT

FIELD OF THE INVENTION

[0001] This invention relates to the field of document management, and in particular to a method and system for delaying the locking of server files being accessed by a user.

BACKGROUND OF THE INVENTION

[0002] Server computers such as web servers can perform double duty. A server can be used as a publishing location, where a user can access content. A server can also be used as a collaboration environment, where two or more users can collaborate on a document. Each user may update or otherwise edit the document. When a server is used as a publishing location, it can deliver an individual copy of a file to any user who requests the file, maintaining the version on the server without changes. When a server is used as a collaboration environment, it can allow a user who wishes to edit the file access to the most recent copy. Often, however, a server is used for both of these purposes at once, supporting both a publishing and a collaboration functions.

[0003] Such a dual-use creates a problem. For example, if the server is a web-server, the default mechanism for accessing a file on a web server (by following a hyper-link) will do only one of two actions. Depending on the hyper-link and web server, either clicking on the hyper-link will create a copy of the file that the user can read without altering the original file or clicking on the hyper-link will open the file on the server and allow the user to edit the file. When a file is opened for editing, it is locked, and no other users are allowed to edit the file.

[0004] Generally, a file may be opened with varying permissions granted to the user opening the file. When opening the file for reading, the file is opened read-only, which allows the user to read/browse or otherwise view or use the data in the file, as long as the data is not changed. When opening for editing, the file is opened read/write, which allows the user the same privileges, but also allows the user to modify the data or append to it. When a file is opened read/write, the file is locked. This locking ensures that only one user is modifying the file at a time, preventing conflicting modifications by two or more users.

[0005] The problem arises when the default action is not what the user desires. For example, if the user wishes to edit the file, if the default is opening a copy for reading, the desired functionality for the user is not provided. Similarly, if the default is opening the file for editing, but the user wished only to read the file, the file is locked and another user who wishes to edit the file will be locked out from editing the file. A user who opens the file for reading and then wishes to edit the file must exit the file and then reopen it for editing.

SUMMARY OF THE INVENTION

[0006] A server is implemented with a modified file open action, which, when a user performs the modified open, initially opens a file without locking the file. When a user indicates (either explicitly or implicitly) that the user is attempting to or intending to open the file, the file can then be locked for editing. In this way, the default action when a user requests a file is to open the file without denying other users access to the file. Then, when the user indicates that editing should occur, the lock for the file is obtained.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The foregoing summary, as well as the following detailed description of preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings exemplary embodiments of the invention; however, the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

[0008] Figure 1 is a block diagram showing an exemplary computing environment in which aspects of the invention may be implemented;

[0009] Figure 2 is a functional diagram illustrating an example of a relationship between a server computer, a network, a user computer, and other computers;

[0010] Figure 3 is a flowchart showing delayed locking in accordance with one embodiment of the present invention; and

[0011] Figure 4 is a flowchart showing delayed locking in accordance with an embodiment of the invention.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

Overview

[0012] In order to allow flexibility when a user opens a document on a shared server, a modified open is implemented. When the modified open is used to open a file, the file is opened read-only. Only when and if the user that has opened the file using the modified open indicates that an edit to the file should be made is the lock on the file obtained, making the file read/write for the user, and locking any other users out of obtaining editing privileges for that file.

Exemplary Computing Device

[0013] FIG. 1 and the following discussion are intended to provide a brief general description of a suitable computing environment in which the invention may be implemented. It should be understood, however, that handheld, portable and other computing devices and computing objects of all kinds are contemplated for use in connection with the present invention, as described above. Thus, while a general purpose computer is described below, this is but one example, and the present invention may be implemented with other computing devices, such as a thin-client having network/bus interoperability and interaction. Thus, the present invention may be implemented in an environment of networked hosted services in which very little or minimal client resources are implicated, e.g., a networked environment in which the client device serves merely as an interface to the network/bus, such as an object placed in an appliance, or other computing devices and objects as well. In essence, anywhere that data may be stored or from which data may be retrieved is a desirable, or suitable, environment for operation according to the invention.

[0014] Although not required, the invention can be implemented via an operating system, for use by a developer of services for a device or object, and/or included within application software that operates according to the invention. Software may be described in the general context of computer-executable instructions, such as program modules, being executed by one or more computers, such as client workstations, servers or other devices. Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer configurations. Other well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but

are not limited to, personal computers (PCs), automated teller machines, server computers, hand-held or laptop devices, multi-processor systems, microprocessor-based systems, programmable consumer electronics, network PCs, appliances, lights, environmental control elements, minicomputers, mainframe computers and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network/bus or other data transmission medium. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices, and client nodes may in turn behave as server nodes.

[0015] Fig. 1 thus illustrates an example of a suitable computing system environment 100 in which the invention may be implemented, although as made clear above, the computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

[0016] With reference to Fig. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer system 110. Components of computer system 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus (also known as Mezzanine bus).

[0017] Computer system 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer system 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, Random Access Memory (RAM),

Read Only Memory (ROM), Electrically Erasable Programmable Read Only Memory (EEPROM), flash memory or other memory technology, Compact Disk Read Only Memory (CDROM), digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer system 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

[0018] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer system 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Fig. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[0019] The computer system 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Fig. 1 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156, such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0020] The drives and their associated computer storage media discussed above and illustrated in Fig. 1 provide storage of computer readable instructions, data structures, program modules and other data for the computer system 110. In Fig. 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer system 110 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus 121, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A graphics interface 182, such as Northbridge, may also be connected to the system bus 121. Northbridge is a chipset that communicates with the CPU, or host processing unit 120, and assumes responsibility for accelerated graphics port (AGP) communications. One or more graphics processing units (GPUs) 184 may communicate with graphics interface 182. In this regard, GPUs 184 generally include on-chip memory storage, such as register storage and GPUs 184 communicate with a video memory 186. GPUs 184, however, are but one example of a coprocessor and thus a variety of coprocessing devices may be included in computer system 110. A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190, which may in turn communicate with video memory 186. In addition to monitor 191, computer systems may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 195.

[0021] The computer system 110 may operate in a networked or distributed environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer system 110, although only a memory storage device 181 has been illustrated in Fig. 1. The logical connections depicted in Fig. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks/buses. Such networking environments are commonplace in homes,

offices, enterprise-wide computer networks, intranets and the Internet.

[0022] When used in a LAN networking environment, the computer system 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer system 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer system 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Fig. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0023] Various distributed computing frameworks have been and are being developed in light of the convergence of personal computing and the Internet. Individuals and business users alike are provided with a seamlessly interoperable and Web-enabled interface for applications and computing devices, making computing activities increasingly Web browser or network-oriented.

[0024] For example, MICROSOFT®'s .NET platform includes servers, building-block services, such as Web-based data storage and downloadable device software. While exemplary embodiments herein are described in connection with software residing on a computing device, one or more portions of the invention may also be implemented via an operating system, application programming interface (API) or a "middle man" object between any of a coprocessor, a display device and requesting object, such that operation according to the invention may be performed by, supported in or accessed via all of .NET's languages and services, and in other distributed computing frameworks as well.

Delaying Locking

[0025] With reference to Figure 2, a file 210 resides on a server computer 200. A user with permissions for makes a request to open the file. The user may be connected to the server computer in a number of ways. For example, the user may connect directly, via a local area network (LAN), or remotely via a network such as the Internet. In the exemplary Figure 2, user computer 220 is connected via network 230 to the server computer 200.

[0026] The user requests that the file be opened. One possible method of submitting such a request is by selecting a hyperlink on a web page served by the server computer 200.

When the request is received by the server, the server allows the user computer 220 to open the file. The file is opened on the user's computer by an application 240. For example, this application may be a word processing application, a spreadsheet application, an audio player application, or any application by which a user may utilize the data being downloaded. In one embodiment, the user selects a hyperlink in a webpage being viewed in a web browsing application on the user computer 220, and the web browsing application opens the file with the appropriate application.

[0027] When the file is opened, in one embodiment, a copy of the file is made in the user computer 230's cache. It is the copy on the server computer 200 which the user browses or reads (or otherwise interacts with – e.g. by playing it in an audio player if the file includes audio data.) When the file is initially opened, the file is opened with “read-only” permissions. In this way more than one user may access the file and use it.

[0028] In an alternate embodiment, instead of a local copy being made in user computer 230, all users with read-only permissions are viewing the file on the server.

[0029] Locking of the file only occurs upon an indication that the user intends to edit the file. The lock would allow the user to have “read/write” permissions on the file. The user could then modify or append file data.

[0030] This intent to edit indication may be one of a number of different things. In one embodiment, the intent to edit is indicated by the user modifying the file in any way. In another embodiment, the intent to edit is indicated by the user. This may occur through a specialized “edit” command incorporated into the application, or when the user requests that changes already made be saved. In another embodiment, the intent to edit determination is a determination that the user has had said file open for a pre-specified period of time. For example, if the user has had the file open in read-only mode for ten minutes, it may be inferred that the user intends to edit the file.

[0031] In one embodiment, when the lock is obtained, a determination is made whether the copy of the file in the user computer 230's cache is the same version as the version currently on the server. This is to ensure that edits have not occurred in the interim. For example, User A opens a first version of the file, but does not demonstrate an intent to edit (and therefore does not trigger a lock). If User A then triggers the locking by demonstrating an intent to edit, it must be ensured that the version of the file currently on the server is the same as that which User A is viewing and intending to edit. If User B has opened and modified the file on the server computer 200 at some point between User A's opening of the file and User A's demonstration of an intent to edit, a conflict may occur. Conflict resolution options should be presented to the user.

[0032] For example, User A may have made substantial changes to the file, and may decide that the changes should supplant any changes that have occurred in the interim. The version on the user computer 230 is then written to the server computer 200 as file 210, and the file is locked so that User A has read/write privileges. Alternatively, User A may decide to save the version that User A is working on locally as another file, and reopen the file 210 on the server computer 200 for editing. Other conflict resolution options may also be presented to the user.

[0033] In order to determine whether the copy of the file in user computer 230 is the same version as the version of the file 210 currently on the server, version information may be stored and compared. This may include a version number, time and date saved information, or other version data. Alternatively, a full comparison may be made of the actual file 210 and the cached version in the user computer 230 to determine if the contents of the file are the same.

[0034] In an alternate embodiment where a user does not copy file 210 to user computer 230 but instead opens it directly from server computer 200, when the intent to edit triggering event occurs, the lock on editing file 210 is obtained for the user.

[0035] As shown in Figure 3, a request is received to open the file, in step 300. In step 310, the file is opened. In one embodiment, the file is opened directly from the server in read-only mode; in an alternate embodiment, a copy of the file is made and opened. In step 320, the determination is made that an editing trigger event has occurred. This editing trigger event may be the user saving changes to a local copy of the file, where one has been made. This editing trigger event may also be the user attempting to change the file, where the server copy of the file was opened read-only. This editing trigger event may also be that a predetermined period of time has elapsed, an "edit" indication received from a user, or any other event which necessitates the locking of the server file for editing.

[0036] In step 330, the lock is obtained on the file. This locks the server copy of the file for the user to modify without the possibility of conflicting modifications.

[0037] As shown in Figure 4, conflict resolution may be implemented in one embodiment. As in Figure 3, the first step 410 is when a request is received to open the file. In the second step 420 a local copy is made of the file, and that local copy is opened for editing. In the third step, 430, a determination is made that an editing trigger event has occurred. A comparison 440 is then made to determine whether the version of the file on the server is the same as the version which had been copied in step 420. If it is the same version, then the lock is obtained on the file, step 450. If the user has made changes to the local copy of the file, the server copy of the file may be updated to reflect those changes. If the version has changed,

conflict resolution options are presented to the user in step 460. This may lead to the lock being obtained on the file, step 450, however, it may also lead to other conflict resolution options.

[0038] In this way, a file may be opened by a number of users simultaneously, who can all use the content of the file. However, when a user wishes to modify the file, the lock is then obtained for the user, and the user will have read/write privileges for the file.

Conclusion

[0039] As mentioned above, while exemplary embodiments of the present invention have been described in connection with various computing devices and network architectures, the underlying concepts may be applied to any computing device or system in which it is desirable to implement a modified file-open. Thus, the methods and systems of the present invention may be applied to a variety of applications and devices. While exemplary programming languages, names and examples are chosen herein as representative of various choices, these languages, names and examples are not intended to be limiting. One of ordinary skill in the art will appreciate that there are numerous ways of providing object code that achieves the same, similar or equivalent systems and methods achieved by the invention.

[0040] The various techniques described herein may be implemented in connection with hardware or software or, where appropriate, with a combination of both. Thus, the methods and apparatus of the present invention, or certain aspects or portions thereof, may take the form of program code (i.e., instructions) embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other machine-readable storage medium, wherein, when the program code is loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for practicing the invention. In the case of program code execution on programmable computers, the computing device will generally include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. One or more programs that may utilize the signal processing services of the present invention, e.g., through the use of a data processing API or the like, are preferably implemented in a high level procedural or object oriented programming language to communicate with a computer. However, the program(s) can be implemented in assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language, and combined with hardware implementations.

[0041] The methods and apparatus of the present invention may also be practiced via communications embodied in the form of program code that is transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via any

other form of transmission, wherein, when the program code is received and loaded into and executed by a machine, such as an EPROM, a gate array, a programmable logic device (PLD), a client computer, a video recorder or the like, or a receiving machine having the signal processing capabilities as described in exemplary embodiments above becomes an apparatus for practicing the invention. When implemented on a general-purpose processor, the program code combines with the processor to provide a unique apparatus that operates to invoke the functionality of the present invention. Additionally, any storage techniques used in connection with the present invention may invariably be a combination of hardware and software.

[0042] While the present invention has been described in connection with the preferred embodiments of the various figures, it is to be understood that other similar embodiments may be used or modifications and additions may be made to the described embodiment for performing the same function of the present invention without deviating therefrom. Furthermore, it should be emphasized that a variety of computer platforms, including handheld device operating systems and other application specific operating systems are contemplated, especially as the number of wireless networked devices continues to proliferate. Therefore, the present invention should not be limited to any single embodiment, but rather should be construed in breadth and scope in accordance with the appended claims.

What is claimed:

- 1. A method for delaying locking of a file being opened, comprising:
receiving a request to open said file;
opening said file for reading;
determining that an editing trigger event has occurred; and
obtaining a lock on said file.**
- 2. The method of claim 1, where said opening of said file comprises opening the file in read-only mode, and where said obtaining a lock comprises changing mode to read/write for said user.**
- 3. The method of claim 1, where said determination that an editing trigger event has occurred comprises receiving a change to said file.**
- 4. The method of claim 1, where said determination that an editing trigger event has occurred comprises receiving a request to edit said file from a user.**
- 5. The method of claim 1, where said determination that an editing trigger event has occurred comprises a determination that a user has had said file open for a pre-specified period of time.**
- 6. The method of claim 1, where said step of opening said file comprises creating a user copy of said file.**
- 7. The method of claim 6, where said file is located on a server computer, a user is using a user computer, said server computer and said user computer are operably connected, and said user copy of said file is local to said user computer.**
- 8. The method of claim 6, where said step of obtaining a lock on said file comprises:
determining if said file has been changed since said user copy was made; and
if said file has not been changed, obtaining a lock on said file.**

9. The method of claim 8, where said step of determining if said file has been changed comprises comparing version information of a current version of said file on said server computer with version information of said user copy.
10. The method of claim 9, where said version information for a specified version of said file comprises information regarding the time of the last change to said specified version.
11. The method of claim 8, where said step of obtaining a lock on said file further comprises: if said file has been changed, offering one or more conflict resolution options to said user.
12. The method of claim 8, where said conflict resolution options comprise the option of saving the user copy of said file.
13. The method of claim 8, where said conflict resolution options comprise the option of retrieving said current version of said file on said server computer.
14. At least one of an operating system, a computer readable medium having stored thereon a plurality of computer-executable instructions, a co-processing device, a computing device, and a modulated data signal carrying computer executable instructions for performing the method of claim 1.
15. A computer system for requesting a file from a server, said server operably connected to said computer system, comprising:
 - an input module for receiving a request to open a file on a server computer;
 - a read-only mode opening module for opening said file in read-only mode, operably connected to said user input module;
 - an editing trigger indication module for determining if an editing trigger has occurred;
 - a read-write mode opening module for opening said file in read-write mode, operably connected to said intent-to-edit indication module; and
 - a file use module for using said file, operably connected to said read-only mode opening module and said read-write mode opening module.
16. The system of claim 15, where said operable connection comprises a network connection.

17. The system of claim 15, where said editing trigger indication module determines if said user has requested a change to said file.
18. The system of claim 15, where said editing trigger indication module determines if said user has requested that said file be made available for editing.
19. The system of claim 15, where said editing trigger indication module determines if said user has had said file open for a pre-specified period of time.
20. The system of claim 15, further comprising:
storage for storing a user copy of said file.
21. The system of claim 20, further comprising:
a change detection module for detecting whether said user copy of said file is a copy of a current version of said file on said server, operably connected to said read-write mode opening module.
22. The system of claim 21, where said read-write mode opening module opens said file for said user in read-write mode if no change in said current version of said file on said server has been detected by said change detection module.
23. The system of claim 21, further comprising:
a change resolution module for presenting at least one change resolution option to said user upon detection of a change in current version of said file on said server.
24. The system of claim 23, where said conflict resolution options comprise the option of saving the user copy of said file.
25. The system of claim 23, where said conflict resolution options comprise the option of retrieving said current version of said file on said server computer.
26. The system of claim 20, where said change detection module compares version information of a current version of said file on said server computer with version information of said user copy.

27. The system of claim 26, where said version information for a specified version of said file comprises information regarding the time of the last change to said specified version.
28. A method for using a remote file on a remote server, comprising:
upon indication from a user that opening of said remote file is requested, receiving a copy of the current version of said remote file from said remote server;
displaying said copy for said user using a software application adapted for use with files of the same type as said remote file;
upon indication from said user that said user intends to edit said remote file, requesting that said file on said remote server be locked for writing by other users.
29. The method of claim 28, where said indication from a user that opening of said remote file is requested comprises said user clicking on a hyperlink.
30. The method of claim 28, where said indication that said user intends to edit said remote file comprises receiving a change to said copy in said application.
31. The method of claim 28, where said indication that said user intends to edit said remote file comprises receiving a request to edit said copy from said user in said application.
32. The method of claim 28, where said indication that said user intends to edit said remote file comprises a determination that said user has had said copy open for a pre-specified period of time.
33. The method of claim 28, where said step of requesting that said file on said remote server be locked for writing by other users comprises:
determining if said remote file has been changed since said copy was made; and
if said remote file has not been changed, requesting that said remote file on said remote server be locked for writing by other users.
34. The method of claim 33, where said step of determining if said file has been changed comprises comparing version information of said remote file on said remote server with version information of said copy.

35. The method of claim 33, where said step of obtaining a lock for said user further comprises:

if said remote file has been changed, offering one or more conflict resolution options to said user.

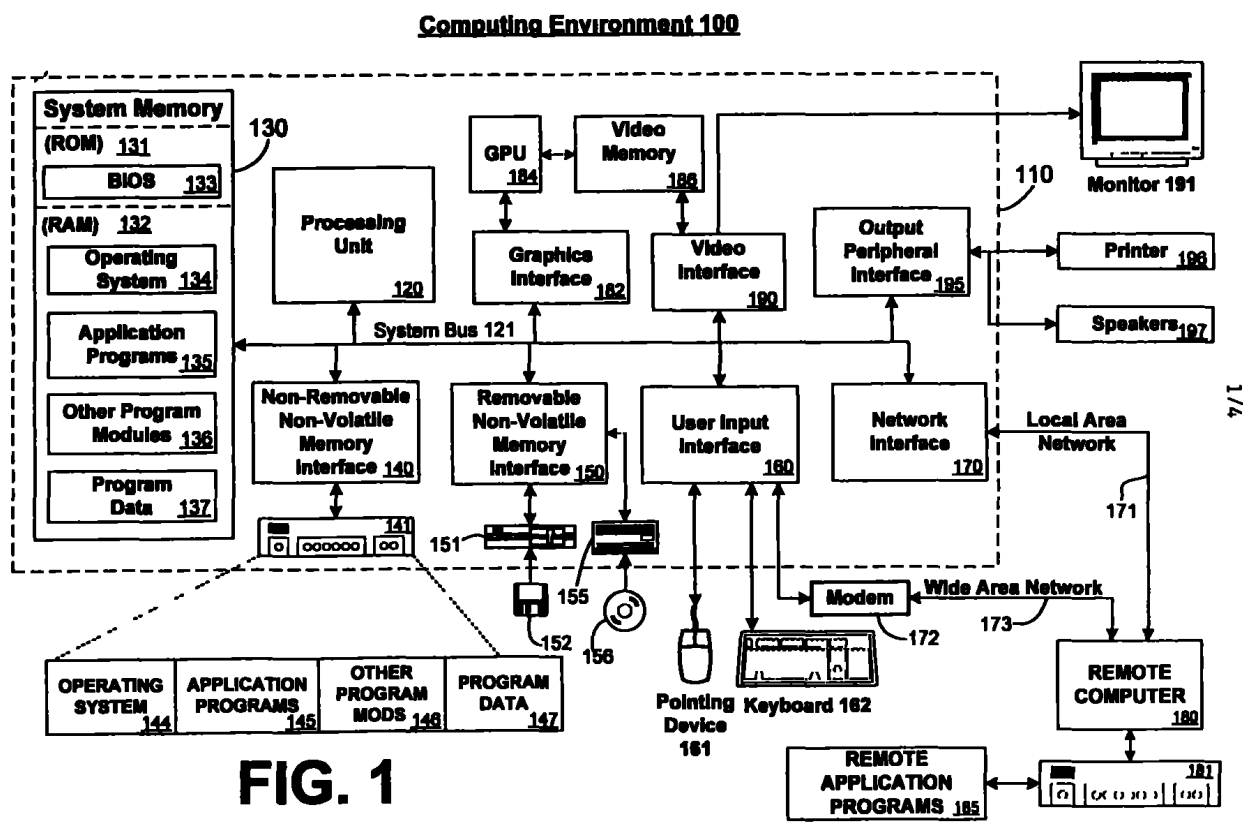
36. At least one of an operating system, a computer readable medium having stored thereon a plurality of computer-executable instructions, a co-processing device, a computing device, and a modulated data signal carrying computer executable instructions for performing the method of claim 28.

Abstract

A server is implemented with a modified file open action, which, when a user performs the modified open, initially opens a file without locking the file. When a user indicates (either explicitly or implicitly) that the user is attempting to or intending to open the file, the file can then be locked for editing. In this way, the default action when a user requests a file is to open the file without denying other users access to the file. Then, when the user indicates that editing should occur, the lock for the file is obtained.

28

68



7-245

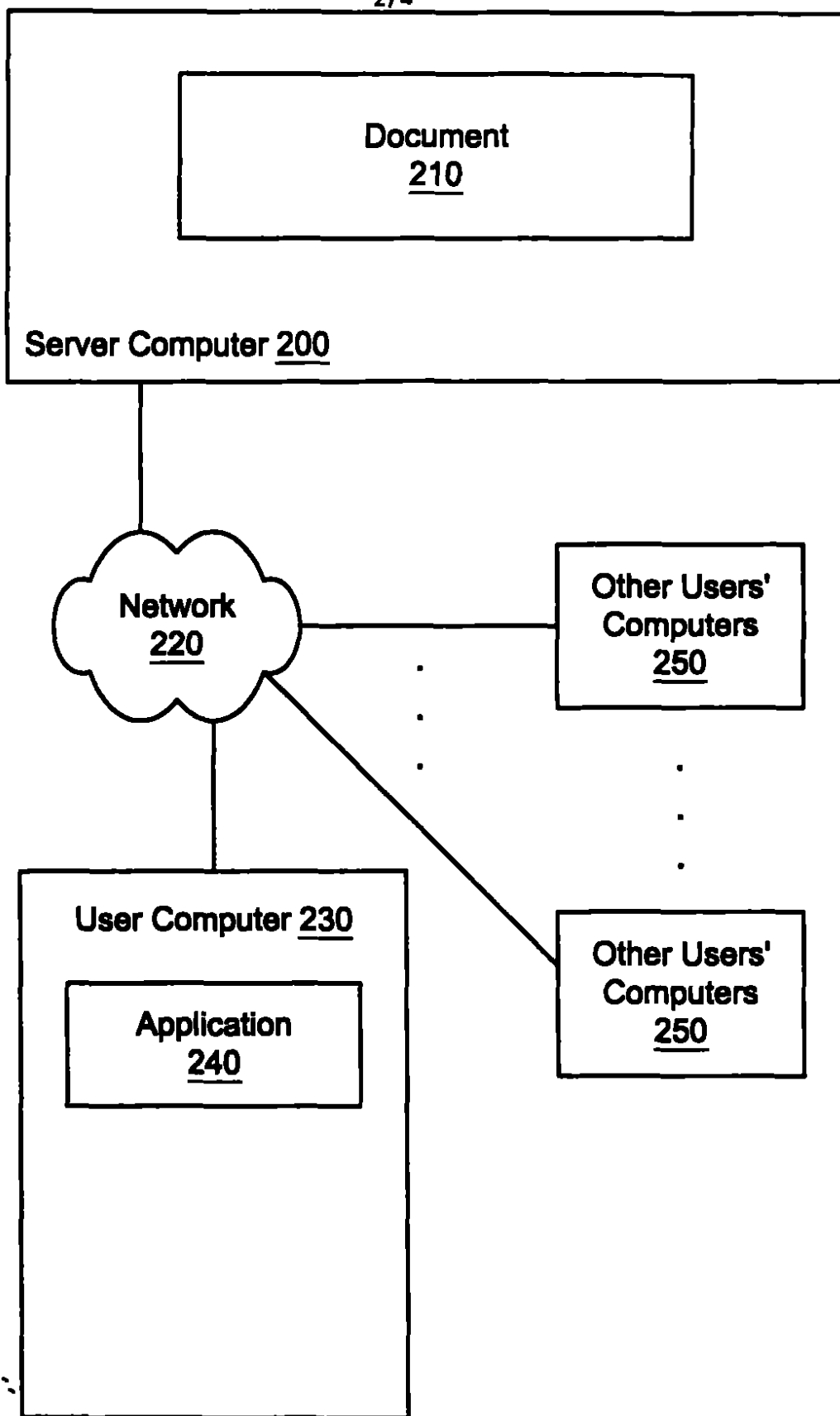


Fig. 2

OK

3/4

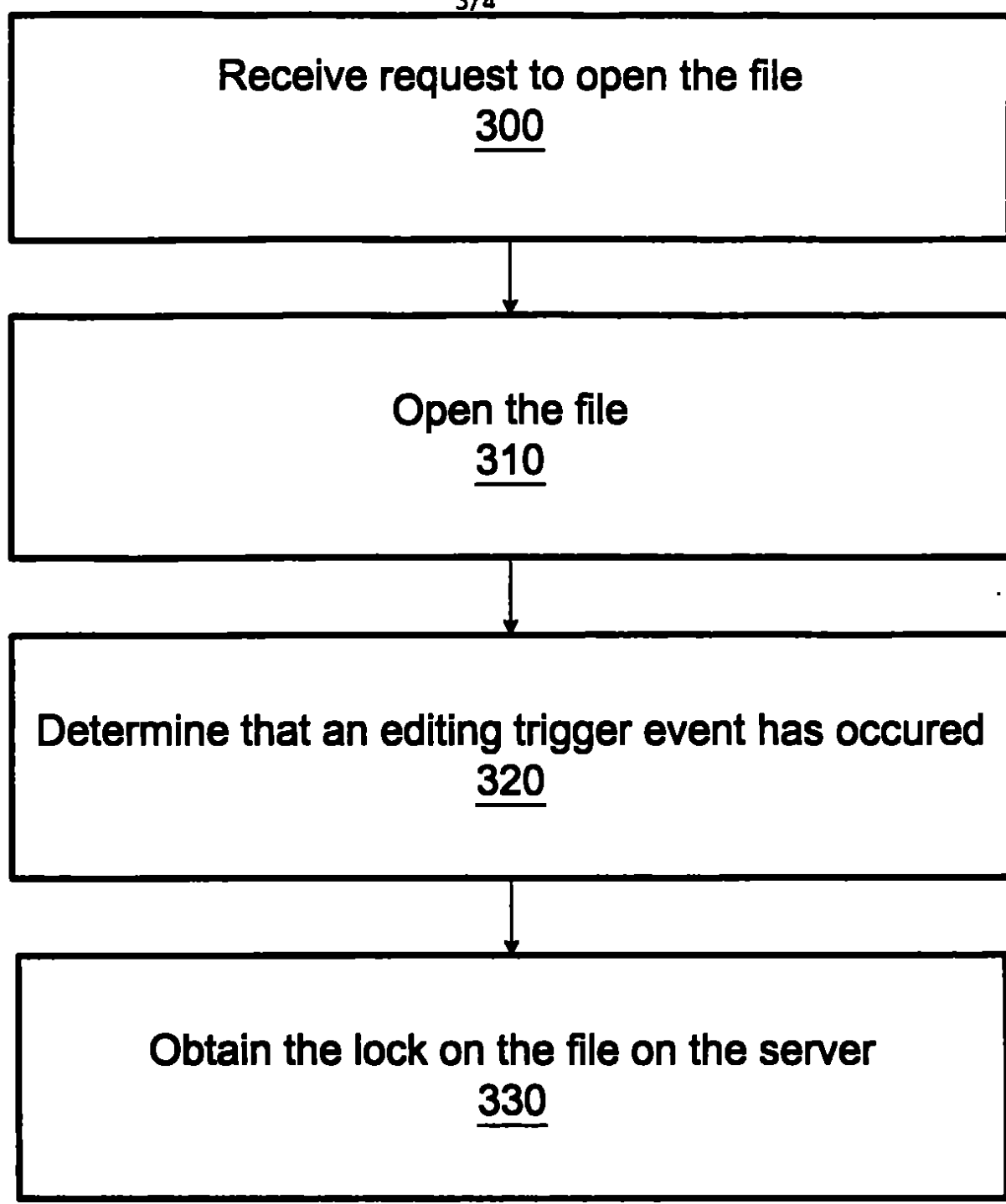


FIG. 3

OK

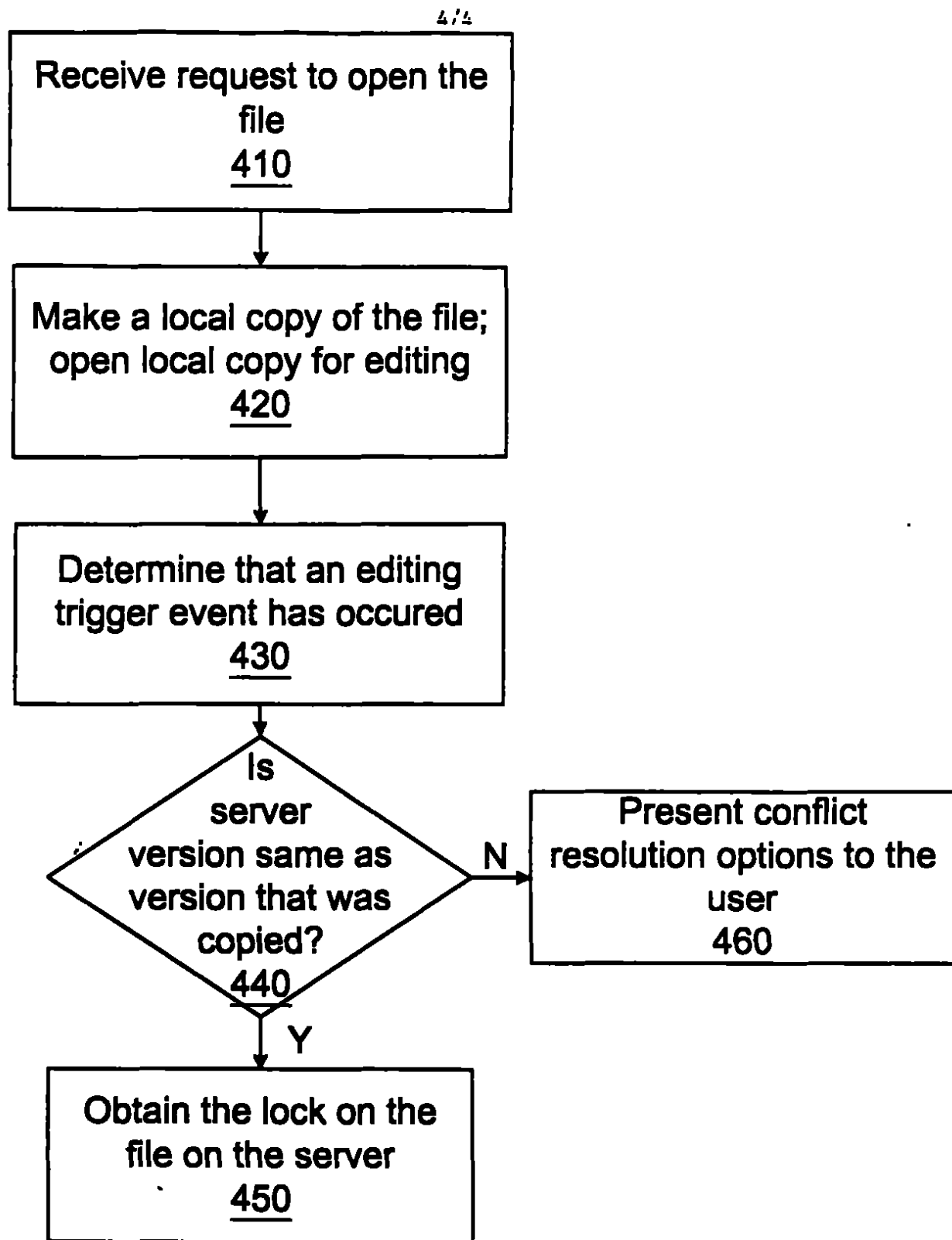


FIG. 4

rt

REQUISITOS NECESARIOS PARA PRESENTACION Y AGILIZACION DEL TRAMITE

	USUARIO	RADICADOR
PATENTE DE INVENCION	()	(/)
MODELO DE UTILIDAD	()	(/)
- Indicación de que se solicita la concesión de una patente.	()	(/)
- Datos de identificación del solicitante o de la persona que se presenta la solicitud.	()	(/)
- Descripción de la invención.	()	(/)
- Dibujos de ser estos pertinentes.	()	(/)
- Comprobante de Pago de las tasas establecidas.	()	(/)
COMPLETA ()	INCOMPLETA ()	()
DISEÑO INDUSTRIAL ()		
- Indicación de que se solicita el registro de Diseño Industrial	()	()
- Datos de identificación del solicitante o de la persona que presenta la solicitud.	()	()
- Representación gráfica y fotográfica del Diseño Industrial o muestra del Material que incorpora el diseño.	()	()
- Comprobante de pago de las tasas establecidas	()	()
COMPLETA ()	INCOMPLETA ()	()
ESQUEMA DE TRAZADO ()		
- Indicación de que solicita el registro de un Esquema de trazado.	()	()
- Datos de identificación del solicitante o de la persona que presenta La solicitud.	()	()
- Representación gráfica del esquema de trazado	()	()
- Comprobante de pago de las tasas establecidas.	()	()
COMPLETA ()	INCOMPLETA ()	()

Firma Responsable:

Claudia V. Nava

Fecha:



13

Industria y Comercio
SUPERINTENDENCIA

PROTOCOLO

Expediente 04-16521

SUPERINDUSTRIA Y COMERCIO SISTEMA DE REGISTRO

MANTENIMIENTO DE PROTOCOLOS

Numero : 16069

Fecha : 04/08/1995

Conferido: MICROSOFT CORPORATION

Pais : US ESTADOS UNIDOS DE AMERICA

Ciudad : 234 REDMOND, WASHINGTON

Represent: S Sustituir: S Desistir: S

No. Radicacion: 95 34920 Clase: 0 Seev: 0 Seac: 0

scnciaCodigo Ctr

Nombre

1 77 A CARDENAS NAVAS DARIO

2 3653 CARDENAS CABALLERO EDUARDO

3 5048 CARDENAS MARTINEZ BERNARDO

Encontrados (1/1) registro(s)

Carrera 13 No. 27-00 Pls 5, 7 y 10

Conmutador: 382 0840

Fax: 350 5220

E-mail: info@sic.gov.co

www.sic.gov.co

Bogotá, D.C., Colombia



74

Industria y Comercio
SUPERINTENDENCIA

Folio 74

2020
Bogotá, D. C.Doctor(a)
DARIO CARDENAS NAVAS
Apoderado(a)

Oficio No.

03416

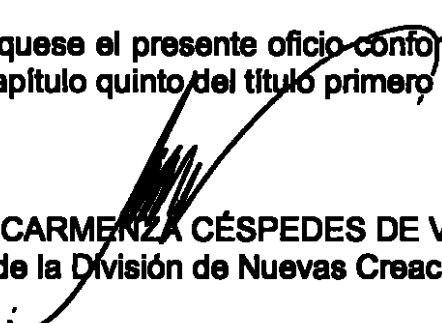
REFERENCIA:	Radicación	04-16521
	Trámite	002
	Evento	001
	Actuación	430
	Folios	001

Como resultado del examen de forma, realizado con fundamento en el artículo 38 de la Decisión 486 de la Comisión de la Comunidad Andina, se le comunica que:

- Allegar copia del documento en que conste la cesión del derecho a la patente otorgado por el inventor al solicitante o a su causante. (art. 26-k, Dec 486/2000).

En cumplimiento del artículo 39 de la Decisión 486, se le requiere para que dentro del término de dos (2) meses siguientes a la fecha de notificación del presente oficio, complete los requisitos anteriormente enunciados. En caso de no dar cumplimiento al presente requerimiento, la solicitud se considerará abandonada y perderá su prelación.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 5.2, literal e), del capítulo quinto del título primero de la Circular Unica No.10 de 2001.



ALIX CARMENZA CÉSPEDES DE VERGEL
Jefe de la División de Nuevas Creaciones

El anterior requerimiento fue notificado por fijación en lista de fecha

01 ABR 2004

Se recuerda al interesado que debe presentar, dentro de los dieciséis meses siguientes contados a partir de la fecha de presentación de la solicitud cuya prioridad se invoca, copia de la misma certificada por la autoridad que la expidió y, en los casos en que haya lugar, la traducción simple del capítulo reivindicatorio. El incumplimiento de este plazo, acarreará la pérdida de la prioridad invocada.

202027

Carrera 13 No. 27-00 Pisos 5, 7 y 10
Conmutador: 382 0840
Fax: 350 5220
E-mail: info@slc.gov.co
www.slc.gov.co
Bogotá, D.C., Colombia