(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 659 872**

(51) Int. CI.:

G06F 9/48 (2006.01)
G06F 9/54 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **12.06.2009 PCT/US2009/047275**

(87) Fecha y número de publicación internacional: **30.12.2009 WO09158220**

(96) Fecha de presentación y número de la solicitud europea: **12.06.2009 E 09770754 (1)**

(97) Fecha y número de publicación de la concesión europea: **27.12.2017 EP 2316091**

(54) Título: **Programación de operaciones en modo protegido**

(30) Prioridad:

27.06.2008 US 163726

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
19.03.2018

(73) Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC (100.0%)
One Microsoft Way
Redmond, WA 98052, US

(72) Inventor/es:

PAPAEFSTATHIOU, EFSTATHIOS;
YU, JINSONG y
OKS, STANISLAV, A.

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Programación de operaciones en modo protegido

Antecedentes

1. Antecedentes y técnica relevante

Los sistemas informáticos y la tecnología relacionada afectan a muchos aspectos de la sociedad. De hecho, la capacidad del sistema informático de procesar información ha transformado la forma en la que vivimos y trabajamos. En la actualidad, los sistemas informáticos realizan comúnmente una gran cantidad de tareas (por ejemplo, procesamiento de textos, gestión de bases de datos, contabilización, etc.) que, antes de la aparición de los sistemas informáticos, se realizaban de forma manual. Más recientemente, los sistemas informáticos se han acoplado entre sí y con otros dispositivos electrónicos para formar redes informáticas tanto por cable como inalámbricas a través de las cuales los sistemas informáticos y otros dispositivos electrónicos pueden transferir datos electrónicos. Por consiguiente, la realización de muchas tareas informáticas se distribuye a lo largo de un número de sistemas informáticos diferentes y / o un número de componentes informáticos diferentes.

Por lo general, cuando un sistema informático se activa o se "arranca" de otro modo, un sistema básico de entrada / salida ("BIOS", *basic input / output system*) se ejecuta a partir de una memoria de solo lectura (por ejemplo, un chip de memoria flash). El BIOS realiza una secuencia de actividades para preparar el sistema informático para la operación. La secuencia de actividades puede incluir (dependiendo de si la misma es un arranque en frío o un reinicio) comprobar la configuración de CMOS para ajustes personalizados, cargar manejadores de interrupciones y controladores de dispositivo, inicializar registros y gestión de alimentación, someter a prueba diversos componentes de soporte físico del sistema informático para asegurar que los mismos están trabajando de forma apropiada (una prueba automática de encendido o "POST", *power on self test*), y activando chips de BIOS en otras tarjetas (por ejemplo, tarjetas gráficas y de SCSI). A partir de los ajustes de CMOS, el BIOS puede identificar dispositivos de arranque e iniciar una secuencia de arranque para iniciar un sistema operativo.

Durante el funcionamiento del sistema informático, el BIOS también proporciona un conjunto de rutinas de bajo nivel que puede usar el sistema operativo para la interconexión con dispositivos de soporte físico diferentes, tales como, por ejemplo, teclado, ratón, pantalla de vídeo, puertos, etc. Por lo tanto, para realizar una tarea informática típica, un sistema operativo recibe comandos (o bien a partir de un programa de aplicación o bien a partir de un usuario) y reenvía esos comandos a unos recursos físicos apropiados. Los recursos físicos, a su vez, implementan operaciones de un nivel más bajo para realizar la tarea informática.

Por lo general, los sistemas operativos incluyen dos modos operativos distintos: un modo de supervisor (al que se hace referencia a veces como "modo de núcleo") y un modo protegido (al que se hace referencia a veces como "modo de usuario"). Cuando se opera en el modo de supervisor, un proceso tiene un acceso sin restricciones a todos los recursos, puede ejecutar cualquier instrucción, y puede hacer referencia a cualquier ubicación de memoria. Por otro lado, cuando se opera en el modo protegido, se restringe el acceso a recursos, se prohíbe la ejecución de algunas instrucciones y se prohíbe hacer referencia a algunas ubicaciones de memoria. Por lo general, los sistemas operativos tienen un componente central o "núcleo" en el que se confía para operar en el modo de núcleo. No se confía en tras porciones del sistema operativo ni como en otros programas, y no se da a los mismos acceso directo a instrucciones y recursos privilegiados. Por lo tanto, un soporte lógico que opera en el modo protegido ha de solicitar el uso del núcleo (por ejemplo, a través de una llamada de sistema) para realizar operaciones privilegiadas.

En general, el sistema operativo gestiona recursos de soporte físico y de soporte lógico de tal modo que el sistema informático en el que reside el mismo se comporta de una forma flexible pero predecible. En un sistema informático típico, estos recursos pueden incluir un procesador, memoria, espacio en disco, etc. El sistema operativo también proporciona una aplicación consistente para invocar otro soporte lógico, tal como, por ejemplo, programas de aplicación. Por lo general, cuando un programa de aplicación ejecuta o crea uno o más procesos para realizar diversas tareas informáticas. Además, la mayor parte de los sistemas operativos permiten que múltiples aplicaciones independientes se ejecuten en un sistema informático al mismo tiempo. Por lo tanto, en cualquier instante dado, un sistema informático puede tener múltiples procesos que se corresponden con múltiples aplicaciones diferentes en ejecución al mismo tiempo.

La creación de procesos es una operación privilegiada que requiere que un programa de aplicación llame al núcleo para realizar la implementación. El núcleo puede asignar memoria para un proceso, cargar código de programa a partir de un disco, y comenzar a ejecutar el proceso.

Cada uno de los múltiples procesos creados puede requerir que uno o más de procesador (CPU), entrada / salida, memoria y recursos de almacenamiento realicen tareas informáticas designadas. No obstante, algunos recursos pueden realizar tareas solo para un número pequeño y limitado de procesos, y a menudo solo un proceso, de una sola vez. Por ejemplo, muchas CPU se limitan a ejecutar instrucciones para un proceso de una sola vez. Por lo tanto, muchos sistemas operativos usan multitarea para dar la apariencia de realizar múltiples tareas informáticas al mismo tiempo. La multitarea es un mecanismo por medio del cual múltiples procesos comparten recursos de procesamiento comunes, tales como, por ejemplo, una CPU.

En algunos sistemas operativos, un proceso puede incluir muchos procesos secundarios a los que se hace referencia a veces como subprocesos de ejecución (o simplemente como “subprocesos”) o fibras. Un subproceso o fibra es un proceso secundario que es una secuencia separada e independiente de ejecución dentro del código de un proceso. Los subprocesos y las fibras proporcionan un mecanismo para que un proceso se ramifique (o se divida) a sí mismo en dos o más tareas que se ejecutan de forma simultánea (o de forma pseudosimultánea). Los subprocesos y las fibras y procesos difieren entre sistemas operativos. No obstante, en general, un subproceso o fibra está contenido en el interior de un proceso y diferentes subprocesos o fibras en el mismo proceso comparten algunos recursos, mientras que procesos diferentes no comparten recursos. Aunque los subprocesos y las fibras pueden solicitar, de forma independiente, acceso a recursos privilegiados si los mismos no tienen ya acceso. Por lo general, los subprocesos y las fibras difieren en que los subprocesos usan multitarea preferente, mientras que las fibras usan multitarea cooperativa.

Por lo general, para implementar una multitarea del tipo deseado, un núcleo de sistema operativo incluye un fragmento de soporte lógico que se denomina programador que determina cuánto tiempo puede emplear ejecutándose cada proceso y / o subproceso (y, en menor medida, las fibras en esos entornos) y en qué orden se ha de pasar el control de la ejecución. El control se pasa a un proceso y / o subproceso por medio del núcleo, que permite el acceso del proceso / subproceso a la CPU y la memoria. En un instante posterior, el control se devuelve al núcleo a través de algún mecanismo para estos otros procesos y / o subprocesos pueden acceder a la CPU y la memoria. Hay una amplia diversidad de algoritmos de programación, tales como, por ejemplo, programación de tiempo virtual prestado (“BVT”, *Borrowed Virtual Time*), programación de colas de realimentación multinivel, programación por intervalos, programación completamente justa, etc. que se pueden implementar en un núcleo de sistema operativo para multiplexar recursos de CPU a múltiples procesos / subprocesos / fibras.

Se hace referencia a veces a pasar el control entre el núcleo y el proceso / subproceso / fibra, y viceversa, como conmutación de contexto. La conmutación de contexto incluye almacenar y restablecer el estado (contexto) de una CPU de tal modo que múltiples procesos pueden compartir un único recurso de CPU. Por lo general, las conmutaciones de contexto son costosas desde el punto de vista del cálculo. Por ejemplo, durante una conmutación de contexto, se detiene un proceso en ejecución y se da una oportunidad de ejecutarse a otro proceso. El núcleo de sistema operativo ha de detener la ejecución del proceso en ejecución, copiar valores en registros de soporte físico en su bloque de control de procesos (“PCB”, *process control block*), actualizar los registros de soporte físico con los valores a partir del PCB del proceso nuevo. Un PCB puede incluir una diversidad de informaciones diferentes dependiendo del sistema operativo. No obstante, en general, un PCB incluye un identificador de proceso (PID, *process identifier*), valores de registro que incluyen un contador de programa para el proceso, el espacio de direcciones del proceso, una prioridad, información de contabilización de proceso (por ejemplo, cuándo se ejecutó por última vez el proceso, cuánto tiempo de CPU ha acumulado el mismo, etc., y un puntero al PCB del siguiente proceso que se va a ejecutar. La totalidad de esta información se ha de descargar para un proceso actual y cargarse para un proceso nuevo en cada conmutación de contexto.

Por consiguiente, en la mayor parte de los sistemas operativos, el núcleo actúa, en esencia, como una autoridad y repositorio centralizado para subprocesos que ejecutan una directiva de todo el sistema. Es decir, el núcleo realiza toda la programación y el mantenimiento de contexto. En ese sentido, el núcleo actúa como cuello de botella, en potencia limitando la escalabilidad. En potencia, también de deteriora el rendimiento cuando un número grande de subprocesos se encuentran activos debido, al menos en parte, a la carga y la descarga sustancial de datos de contexto.

Además, un núcleo de sistema operativo define tanto la abstracción de ejecución (por ejemplo, subprocesos, fibras, etc.) como el algoritmo de programación que va a usar el núcleo. Por lo tanto, en esencia se obliga a todas las aplicaciones que ejecutan el sistema operativo a usar la abstracción de ejecución y el algoritmo de programación definidos. Esto no llega a ser óptimo, debido a que algunas aplicaciones pueden ser más convenientes para su uso con otras abstracciones de ejecución y / o algoritmos de programación. Por ejemplo, una aplicación puede ser muy conveniente para usar tareas pendientes (un tipo de abstracción de ejecución) y programación de colas de realimentación multinivel. No obstante, si el núcleo define subprocesos con programación de BVT, se obliga no obstante a la aplicación a usar esa combinación.

Debido a que muchas aplicaciones tienen diferentes requisitos, en esencia también es imposible obtener un algoritmo de programación único (o de talla única) que sea óptimo para todas las aplicaciones. Por ejemplo, cada uno de un procesador de textos, un reproductor de medios y una aplicación de realización de copia de seguridad de disco puede tener unos algoritmos de programación óptimos diferentes. No obstante, se pueden realizar compensaciones recíprocas para definir un algoritmo de programación que sea al menos funcional, si bien de forma no óptima, a lo largo de una gama de aplicaciones.

El documento US 6.766.515 B1 se refiere a técnicas para programar procesos paralelos sin comunicación alguna de núcleo a núcleo. Un modelo de muchos a muchos multiplexa subprocesos de nivel de usuario en un número más pequeño de procesadores virtuales, a menudo subprocesos de nivel de núcleo. Por lo general, la arquitectura se implementa mediante la construcción de un programador de nivel de usuario que gestiona la conmutación de los subprocesos de nivel de usuario en los subprocesos de nivel de núcleo. Entonces, un programador de núcleo es responsable de programar los procesadores virtuales en procesadores físicos. Una característica de un modelo de

nanosubprocesos es que el núcleo deja de proporcionar programación directa alguna de subprocesos. Más bien, el programador de núcleo asigna procesadores a aplicaciones, entonces un programador de nivel de usuario programa subprocesos de nivel de usuario en los procesadores asignados. Por lo tanto, la abstracción del procesador virtual se sustituye con la abstracción de múltiples procesadores virtuales. En un ejemplo, se describe una técnica de conmutación entre un primer subproceso de usuario y un segundo subproceso de usuario al tiempo que se permanece dentro del espacio de usuario.

Govindan, Ramesh y col.: "*Scheduling and IPC Mechanisms for Continuous Media*", División de Ciencias Informáticas, Departamento de Ingeniería Eléctrica y Ciencias Informáticas, Universidad de California, Berkeley, *Operating Systems Review (SIGOPS)* 25 (1991), n.º 5 describe que las estaciones de trabajo de próxima generación tendrán soporte de soporte físico para "medios continuos" (CM, *continuous media*) digitales tales como audio y vídeo. Las aplicaciones de CM manejan datos a velocidades elevadas, con unos requisitos de temporización estrictos, y a menudo en "fragmentos" pequeños. Si tales aplicaciones se han de ejecutar de una forma eficiente y predecible como programas de nivel de usuario, un sistema operativo ha de proporcionar mecanismos de programación y de IPC que reflejen estas necesidades. Se proponen dos de tales mecanismos: programación de CPU de nivel de división de procesos ligeros en múltiples espacios de direcciones, y secuencias asignadas a la memoria para el movimiento de datos entre espacios de direcciones.

Breve sumario

El objeto de la presente invención es mejorar la eficiencia de un proceso de programación. Este objeto se soluciona por medio de la materia objeto de las reivindicaciones independientes. Las formas de realización preferidas se definen por medio de las reivindicaciones dependientes.

La presente invención se extiende a procedimientos, sistemas y productos de programa informático para la programación de operaciones en modo protegido. En algunas formas de realización, un sistema informático configura la programación de los recursos de procesador para que tengan lugar en el modo de protegido (por ejemplo, de usuario) para descentralizar la responsabilidad de la programación con respecto al modo de supervisor (por ejemplo, de núcleo). El sistema informático crea un dominio de programación de modo protegido que opera en el modo protegido de un sistema operativo. La creación de un dominio de programación de modo protegido incluye crear un procesador virtual. El procesador virtual asigna al menos una porción del procesador físico para su uso por objetos de ejecución del dominio de programación de modo protegido.

El procesador virtual incluye un procesador virtual de modo protegido. El procesador virtual de modo protegido está configurado para procesar notificaciones de modo de supervisor y distribuir objetos de ejecución. El procesador virtual también incluye un procesador virtual de modo de supervisor. El procesador virtual de modo de supervisor está configurado para reservar al menos una porción del procesador físico para su uso por objetos de ejecución del dominio de programación de modo protegido.

La creación de un dominio de programación de modo protegido también incluye cargar al menos una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto. La al menos una directiva de programación de modo protegido es para multiplexar objetos de ejecución del dominio de programación de modo protegido en el procesador virtual. La creación de un dominio de programación de modo protegido también incluye crear un almacén de objetos de ejecución que está configurado para almacenar el estado para objetos de ejecución del dominio de programación de modo protegido. El almacén de objetos de ejecución facilita la conmutación del contexto entre diferentes objetos de ejecución dentro del dominio de programación de modo protegido.

En otras formas de realización, un sistema informático programa el consumo de recursos de procesador en el modo protegido para descentralizar la responsabilidad de la programación con respecto al modo de supervisor. El sistema informático asigna recursos de procesador físicos a un primer objeto de ejecución de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido. Los recursos de procesador físico se asignan de acuerdo con una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto. El sistema informático utiliza los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el primer objeto de ejecución.

El sistema informático determina, de acuerdo con la directiva de programación de modo protegido, que la asignación de recursos de procesador físico va a realizar una transición a un segundo objeto de ejecución dentro del dominio de programación de modo protegido antes de completar completamente el trabajo que se indica en el primer objeto de ejecución. El sistema informático conmuta el contexto desde el primer objeto de ejecución al segundo objeto de ejecución dentro del modo protegido y sin realizar una transición al modo de supervisor.

La conmutación del contexto incluye conservar el estado del primer objeto de ejecución en un almacén de objetos de ejecución residente dentro del dominio de programación de modo protegido. La conmutación del contexto también incluye cargar el estado del segundo objeto de ejecución a partir del almacén de objetos de ejecución. El sistema informático asigna recursos de procesador físico al segundo objeto de ejecución del dominio de programación de modo protegido dentro del dominio de programación de modo protegido posteriormente a la conmutación del

contexto al segundo objeto de ejecución. El sistema informático utiliza los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el segundo objeto de ejecución.

El presente sumario se proporciona para presentar una selección de conceptos en una forma simplificada que se describen adicionalmente en lo sucesivo en la Descripción detallada. El presente Sumario no tiene por objeto identificar características clave o características esenciales de la materia objeto que se reivindica, ni se tiene por objeto que la misma se use como una ayuda para determinar el ámbito de la materia objeto que se reivindica.

Algunas características y ventajas de la invención se expondrán en la descripción que sigue y, en parte, serán obvias a partir de la descripción, o se pueden aprender mediante la práctica de la invención. Las características y ventajas de la invención se pueden lograr y obtener por medio de los instrumentos y las combinaciones que se indican de forma particular en las reivindicaciones adjuntas. Estas y otras características de la presente invención se volverán más plenamente evidentes a partir de la siguiente descripción y las reivindicaciones adjuntas, o se pueden aprender mediante la práctica de la invención tal como se expone en lo sucesivo en el presente documento.

Breve descripción de los dibujos

Con el fin de describir la forma en la que se pueden obtener las ventajas y características indicadas en lo que antecede así como otras ventajas y características de la invención, se presentará una descripción más particular de la invención que se ha descrito en lo que antecede brevemente por referencia a formas de realización específicas de la misma que se ilustran en los dibujos adjuntos. Entendiendo que estos dibujos muestran solo formas de realización típicas de la invención y, por lo tanto, no se ha de considerar que sean limitantes de su ámbito, la invención se describirá y se explicará de forma más específica y con mayor detalle a través del uso de los dibujos adjuntos, en los que:

La figura 1A ilustra una porción de una arquitectura informática a modo de ejemplo que facilita configurar la programación de los recursos de procesador para que tengan lugar en el modo protegido para descentralizar la responsabilidad de la programación con respecto al modo de supervisor.
La figura 1B ilustra una porción de la arquitectura informática a modo de ejemplo que programa el consumo de recursos de procesador en el modo protegido para descentralizar la responsabilidad de la programación con respecto al modo de supervisor.
La figura 1C ilustra una porción de la arquitectura informática a modo de ejemplo que muestra componentes adicionales de un procesador virtual.
La figura 1D ilustra una porción de la arquitectura informática a modo de ejemplo que muestra múltiples procesadores virtuales en un dominio de programación de modo protegido.
La figura 1E ilustra una porción de la arquitectura informática a modo de ejemplo que muestra componentes para sincronizar entre dominios de programación de modo protegido.
Las figuras 2A y 2B ilustran unas infraestructuras de modo de supervisor a modo de ejemplo que proporcionan características a los dominios de programación de modo protegido.
La figura 3 ilustra una arquitectura de programación de modo de usuario a modo de ejemplo.
La figura 4 ilustra un diagrama de flujo de un procedimiento a modo de ejemplo para configurar la programación de los recursos de procesador para que tengan lugar en el modo protegido.
La figura 5 ilustra un diagrama de flujo de un procedimiento a modo de ejemplo para programar recursos de procesador en el modo protegido.

Descripción detallada

La presente invención se extiende a procedimientos, sistemas y productos de programa informático para la programación de operaciones en modo protegido. En algunas formas de realización, un sistema informático configura la programación de los recursos de procesador para que tengan lugar en el modo de protegido (por ejemplo, de usuario) para descentralizar la responsabilidad de la programación con respecto al modo de supervisor (por ejemplo, de núcleo). El sistema informático crea un dominio de programación de modo protegido que opera en el modo protegido de un sistema operativo. La creación de un dominio de programación de modo protegido incluye crear un procesador virtual. El procesador virtual asigna al menos una porción del procesador físico para su uso por objetos de ejecución del dominio de programación de modo protegido.

El procesador virtual incluye un procesador virtual de modo protegido. El procesador virtual de modo protegido está configurado para procesar notificaciones de modo de supervisor y distribuir objetos de ejecución. El procesador virtual también incluye un procesador virtual de modo de supervisor. El procesador virtual de modo de supervisor está configurado para reservar al menos una porción del procesador físico para su uso por objetos de ejecución del dominio de programación de modo protegido.

La creación de un dominio de programación de modo protegido también incluye cargar al menos una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto. La al menos una directiva de programación de modo protegido es para multiplexar objetos de ejecución del dominio de programación de modo protegido en el procesador virtual. La creación de un dominio de programación de modo protegido también incluye crear un almacén de objetos de ejecución que está configurado para almacenar el estado

para objetos de ejecución del dominio de programación de modo protegido. El almacén de objetos de ejecución facilita la conmutación del contexto entre diferentes objetos de ejecución dentro del dominio de programación de modo protegido.

En otras formas de realización, un sistema informático programa el consumo de recursos de procesador en el modo protegido para descentralizar la responsabilidad de la programación con respecto al modo de supervisor. El sistema informático asigna recursos de procesador físico a un primer objeto de ejecución de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido. Los recursos de procesador físico se asignan de acuerdo con una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto. El sistema informático utiliza los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el primer objeto de ejecución.

El sistema informático determina, de acuerdo con la directiva de programación de modo protegido, que la asignación de recursos de procesador físico va a realizar una transición a un segundo objeto de ejecución dentro del dominio de programación de modo protegido antes de completar completamente el trabajo que se indica en el primer objeto de ejecución. El sistema informático conmuta el contexto desde el primer objeto de ejecución al segundo objeto de ejecución dentro del modo protegido y sin realizar una transición al modo de supervisor.

La conmutación del contexto incluye conservar el estado del primer objeto de ejecución en un almacén de objetos de ejecución residente dentro del dominio de programación de modo protegido. La conmutación del contexto también incluye cargar el estado del segundo objeto de ejecución a partir del almacén de objetos de ejecución. El sistema informático asigna recursos de procesador físico al segundo objeto de ejecución del dominio de programación de modo protegido dentro del dominio de programación de modo protegido posteriormente a la conmutación del contexto al segundo objeto de ejecución. El sistema informático utiliza los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el segundo objeto de ejecución.

Las formas de realización de la presente invención pueden comprender o utilizar un ordenador de propósito especial o de propósito general que incluye soporte físico informático, tal como se analiza con mayor detalle en lo sucesivo. Las formas de realización dentro del ámbito de la presente invención también incluyen soportes legibles por ordenador físicos y de otro tipo para portar o almacenar instrucciones ejecutables por ordenador y / o estructuras de datos. Tales soportes legibles por ordenador pueden ser cualesquiera medios disponibles a los que se pueda acceder por medio de un sistema de ordenador de propósito general o de propósito especial. Los soportes legibles por ordenador que almacenan instrucciones ejecutables por ordenador son medios de almacenamiento físicos. Los soportes legibles por ordenador que portan instrucciones ejecutables por ordenador son medios de transmisión. Por lo tanto, a modo de ejemplo, y no de limitación, las formas de realización de la invención pueden comprender al menos dos tipos claramente diferentes de soportes legibles por ordenador: medios de almacenamiento físicos y medios de transmisión.

Los medios de almacenamiento físicos incluyen RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro soporte que se pueda usar para almacenar medios de código de programa deseados en forma de instrucciones ejecutables por ordenador o estructuras de datos y al que se pueda acceder por medio de un ordenador de propósito general o de propósito especial.

Con la presente descripción y las siguientes reivindicaciones, una “red” se define como uno o más enlaces de datos que posibilitan el transporte de datos electrónicos entre sistemas informáticos y / o módulos y / u otros dispositivos electrónicos. Cuando se transfiere o se proporciona información a través de una red u otra conexión de comunicaciones (ya sea cableada, inalámbrica, o una combinación de cableada e inalámbrica) a un ordenador, el ordenador ve de forma apropiada la conexión como un medio de transmisión. Los medios de transmisión pueden incluir una red y / o enlaces de datos que se pueden usar para portar medios de código de programa deseados en forma de instrucciones ejecutables por ordenador o estructuras de datos y a los que se pueda acceder por medio de un ordenador de propósito general o de propósito especial. También se deberían incluir combinaciones de lo anterior dentro del ámbito de los soportes legibles por ordenador.

Además, se debería entender que, tras alcanzar diversos componentes de sistema informático, se pueden transferir medios de código de programa en forma de instrucciones ejecutables por ordenador o estructuras de datos de forma automática de los medios de transmisión a los medios de almacenamiento físicos (o viceversa). Por ejemplo, las instrucciones ejecutables por ordenador o estructuras de datos que se reciben a través de una red o enlace de datos se pueden almacenar de forma temporal en RAM dentro de un módulo de interfaz de red (por ejemplo, un “NIC”) y, entonces, transferirse con el tiempo a una RAM de sistema informático y / o a medios de almacenamiento físicos menos volátiles en un sistema informático. Por lo tanto, se debería entender que se pueden incluir medios de almacenamiento físicos en componentes de sistema informático que también (o incluso principalmente) utilizan medios de transmisión.

Las instrucciones ejecutables por ordenador comprenden, por ejemplo, instrucciones y datos que dan lugar a que un ordenador de propósito general, un ordenador de propósito especial o un dispositivo de procesamiento de propósito especial realice una determinada función o grupo de funciones. Las instrucciones ejecutables por ordenador pueden

ser, por ejemplo, archivos binarios, instrucciones de formato intermedio tales como lenguaje ensamblador, o incluso código fuente. A pesar de que la materia objeto se ha descrito en un lenguaje específico de características estructurales y / o actos metodológicos, se ha de entender que la materia objeto que se define en las reivindicaciones adjuntas no está necesariamente limitada a las características o los actos descritos que se han descrito en lo que antecede. Más bien, las características y los actos descritos se desvelan como formas a modo de ejemplo de implementación de las reivindicaciones.

Los expertos en la materia apreciarán que la invención se puede poner en práctica en entornos informáticos en red con muchos tipos de configuraciones de sistema informático, incluyendo, ordenadores personales, ordenadores de escritorio, ordenadores portátiles, procesadores de mensajes, dispositivos de mano, sistemas de múltiples procesadores, electrónica de consumo programable o basada en microprocesador, PC en red, miniordenadores, ordenadores de gran sistema, teléfonos móviles, PDA, buscapersonas, encaminadores, conmutadores y similares. La invención también se puede poner en práctica en entornos de sistema distribuidos en los que realizan tareas sistemas informáticos tanto locales como remotos, que están vinculados (ya sea por enlaces de datos cableados, enlaces de datos inalámbricos, o por una combinación de enlaces de datos cableados e inalámbricos) a través de una red. En un entorno de sistema distribuido, se pueden ubicar módulos de programa en dispositivos de almacenamiento en memoria tanto locales como remotos.

Dentro de la presente descripción y las siguientes reivindicaciones, un “objeto de ejecución” se define como una instancia de una abstracción de carga de trabajo que consume recursos y se comparte en el tiempo (se multiplexa) en un procesador. Los ejemplos de objetos de ejecución incluyen: subprocesos, tareas, fibras, promesas, tareas pendientes, etc.

Dentro de la presente descripción y las siguientes reivindicaciones, “programador”, “algoritmo de programación”, y “algoritmo de directiva de programación” se definen como un conjunto de componentes para multiplexar objetos de ejecución en un procesador. Los programadores, los algoritmos de programación y los algoritmos de directiva de programación pueden implementar una diversidad de tipos diferentes de programación, incluyendo: programación de tiempo virtual prestado (“BVT”, *Borrowed Virtual Time*), programación completamente justa (“CFS”, *completely fair scheduling*), tipos diferentes de programación de operación por turnos, programación basada en cola (FIFO, multinivel, etc.), programación basada en pila (LIFO), programación de conjunto, programación por intervalos, distribución de turnos, etc.

La figura 1A ilustra una porción de la arquitectura informática 100 que facilita configurar la programación de los recursos de procesador para que tengan lugar en el modo protegido con el fin de descentralizar la responsabilidad de la programación con respecto al modo de supervisor. Tal como se muestra, la arquitectura informática 100 incluye el procesador 109 (y, posiblemente, uno o más procesadores adicionales) y el sistema operativo 101. El sistema operativo 101 puede ejecutar aplicaciones en el modo protegido 102 (por ejemplo, modo de usuario) y el modo de supervisor 103 (por ejemplo, el modo de núcleo). El modo protegido 102 puede limitar el acceso de los programas a los recursos del procesador físico 109. Por otro lado, el modo de supervisor 103 tiene un acceso ilimitado a los recursos del procesador físico 109. Una o más funciones controlan el acceso al modo de supervisor 103 para permitir que un programa en el modo protegido 102 realice una transición al modo de supervisor 103 para los recursos del procesador físico 109.

Por lo tanto, los programas que se ejecutan dentro del sistema operativo 101 pueden realizar una transición desde el modo protegido 102 al modo de supervisor 103 y desde el modo de supervisor 103 al modo protegido 102 dependiendo de instrucciones de programa. El modo de supervisor 103 tiene un formato de objeto de ejecución de modo de supervisor por defecto (por ejemplo, subprocesos) que consume los recursos del procesador físico 109 de una forma compartida en el tiempo de acuerdo con una directiva de programación de modo de supervisor por defecto (por ejemplo, una programación de “BVT”).

La figura 4 ilustra un diagrama de flujo de un procedimiento 400 a modo de ejemplo para configurar la programación de los recursos de procesador para que tengan lugar en el modo protegido. El procedimiento 400 se describirá con respecto a los componentes y datos que se muestran en la figura 1A.

El procedimiento 400 incluye un acto de crear un dominio de programación de modo protegido que opera en el modo protegido del sistema operativo (el acto 401). Por ejemplo, el sistema operativo 101 puede crear el dominio de programación de modo protegido 104 dentro del modo protegido 102.

La creación de un dominio de programación de modo protegido incluye un acto de crear un procesador virtual, asignando el procesador virtual al menos una porción del procesador físico para su uso por objetos de ejecución del dominio de programación de modo protegido (el acto 402). Por ejemplo, el sistema operativo 101 puede crear el procesador virtual 108. En general, el procesador virtual 108 realiza una abstracción del formato de objeto de ejecución por defecto y el programador del modo de supervisor 103 a partir del dominio de programación de modo protegido 104. Por ejemplo, el procesador virtual 108 asigna la asignación de recursos 112 (una porción de los recursos del procesador físico 109) para su uso por objetos de ejecución dentro del dominio de programación de modo de protección 104.

El procesador virtual 108 incluye el procesador virtual de modo protegido 108P. El procesador virtual de modo protegido 108 está configurado para procesar notificaciones de modo de supervisor a partir del modo de supervisor 103 y distribuir objetos de ejecución dentro del dominio de programación de modo protegido 104. El procesador virtual 108 también incluye el procesador virtual de modo de supervisor 108S. El procesador virtual de modo de supervisor 108S está configurado para reservar al menos una porción del procesador físico 109 (por ejemplo, la asignación de recursos 112) para su uso por objetos de ejecución del dominio de programación de modo protegido 104.

La creación de un dominio de programación de modo protegido también incluye cargar al menos una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto, la al menos una directiva de programación de modo protegido para multiplexar objetos de ejecución del dominio de programación de modo protegido en el procesador virtual (el acto 403). Por ejemplo, el sistema operativo 101 puede cargar el algoritmo de directiva de programación 107 para multiplexar objetos de ejecución dentro del dominio de programación de modo protegido 104. El algoritmo de directiva de programación 107 puede ser virtualmente cualquier directiva de programación, tal como, por ejemplo, programación, programación de colas de realimentación multinivel, programación por intervalos, programación completamente justa, etc., que difiere de la directiva de programación por defecto del modo de supervisor 103. El estado de directiva 116 pueden ser estructuras de datos para conservar el estado de directiva para el algoritmo de directiva de programación 107.

La creación de un dominio de programación de modo protegido también incluye crear un almacén de objetos de ejecución que está configurado para almacenar el estado para objetos de ejecución del dominio de programación de modo protegido para facilitar la conmutación del contexto entre diferentes objetos de ejecución dentro del dominio de programación de modo protegido (el acto 404). Por ejemplo, el sistema operativo 101 puede crear el almacén de objetos de ejecución 105 para almacenar el estado para objetos de ejecución del dominio de programación de modo protegido 104 para facilitar la conmutación entre objetos dentro del dominio de programación de modo protegido 104. Los objetos de ejecución del dominio de programación de modo protegido 104 (por ejemplo, tareas pendientes) pueden diferir de los objetos de ejecución que se usan en el modo de supervisor 103 (subprocesos).

Posteriormente a la configuración del dominio de programación de modo protegido 104, los recursos del procesador físico 109 (por ejemplo, la asignación de recursos 112) se pueden multiplexar entre diferentes objetos de ejecución dentro del dominio de programación de modo protegido 104. La figura 1B ilustra una porción de la arquitectura informática 100 a modo de ejemplo que programa el consumo de recursos de procesador en el modo protegido para descentralizar la responsabilidad de la programación con respecto al modo de supervisor.

Tal como se muestra en la figura 1B, el dominio de programación de modo protegido 104 incluye adicionalmente el módulo de sincronización 114. En general, los componentes del dominio de programación de modo protegido 104 permiten que tengan lugar, dentro del proceso, operaciones de programación, tales como, por ejemplo, bloqueo / desbloqueo debido a la sincronización, producción o terminación y conmutación de contexto. El módulo de sincronización 114 se puede configurar para su uso con un tipo especificado de objeto de ejecución. Puede existir una relación intrínseca entre el módulo de sincronización 114 y el algoritmo de directiva de programación 107, debido a que las operaciones de sincronización pueden producir transiciones de estado de objeto de ejecución que afectan al estado de directiva 116. El módulo de sincronización 114 se puede generar / cargar durante o después de la creación del dominio de programación de modo de protección 104, dependiendo, al menos en parte, del tipo de objeto de ejecución que se va a usar.

La figura 5 ilustra un diagrama de flujo de un procedimiento 500 a modo de ejemplo para programar recursos de procesador en el modo protegido. El procedimiento 500 se describirá con respecto a los componentes y datos que se muestran en la figura 1B.

El procedimiento 500 incluye un acto de asignar recursos de procesador físico a un primer objeto de ejecución de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido, recursos de procesador físico que se asignan de acuerdo con una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto (el acto 501). Por ejemplo, el sistema operativo 101 puede asignar la asignación de recursos 112 al objeto de ejecución 131 de acuerdo con el algoritmo de directiva de programación 107. El procedimiento 500 incluye un acto de utilizar los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el primer objeto de ejecución (el acto 502). Por ejemplo, el objeto de ejecución 131 puede utilizar la asignación de recursos 112 para completar parcialmente el trabajo que se indica en el objeto de ejecución 131.

El procedimiento 500 incluye un acto de determinar, de acuerdo con la directiva de programación de modo protegido, que la asignación de recursos de procesador físico va a realizar una transición a un segundo objeto de ejecución dentro del dominio de programación de modo protegido antes de completar completamente el trabajo que se indica en el primer objeto de ejecución

(el acto 503). Por ejemplo, el sistema operativo 101 puede determinar, de acuerdo con el algoritmo de directiva de programación 107, que la asignación de recursos 112 va a realizar una transición del objeto de ejecución 132, antes de completar el trabajo que se indica en el objeto de ejecución 131.

El procedimiento 500 incluye un acto de conmutar el contexto desde el primer objeto de ejecución al segundo objeto de ejecución dentro del modo protegido y sin realizar una transición al modo de supervisor (el acto 504). Por ejemplo, el sistema operativo 101 puede conmutar el contexto desde el objeto de ejecución 131 al objeto de ejecución 132 dentro del dominio de programación de modo protegido 104 y sin conmutar al modo de supervisor 103.

La conmutación del contexto incluye un acto de conservar el estado del primer objeto de ejecución en un almacén de objetos de ejecución, el almacén de objetos de ejecución residente dentro del dominio de programación de modo protegido (el acto 505). Por ejemplo, en el momento de la conmutación de contexto, el objeto de ejecución 131 puede tener un estado de EO 133. El módulo de sincronización 114 puede conservar el estado de EO 133 en el almacén de objetos de ejecución 105. La conmutación del contexto también incluye un acto de cargar el estado del segundo objeto de ejecución a partir del almacén de objetos de ejecución (el acto 506). Por ejemplo, el módulo de sincronización 114 puede cargar el estado de EO 134 a partir del almacén de objetos de ejecución 105. El estado de EO 134 se puede haber conservado previamente cuando se conmutó el contexto lejos del objeto de ejecución 132.

El procedimiento 500 incluye un acto de asignar recursos de procesador físico al segundo objeto de ejecución de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido posteriormente a la conmutación del contexto al segundo objeto de ejecución (el acto 507). Por ejemplo, el sistema operativo 101 puede asignar la asignación de recursos 112 al objeto de ejecución 132 dentro del dominio de programación de modo protegido 104 posteriormente a la conmutación del contexto al objeto de ejecución 132. El procedimiento 500 incluye un acto de utilizar los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el segundo objeto de ejecución (el acto 508). Por ejemplo, la asignación de recursos 112 se puede utilizar para completar parcialmente el trabajo que se indica en el objeto de ejecución 132.

En general, la programación del modo protegido (por ejemplo, de modo de usuario) descentraliza las operaciones de programación con respecto al modo de supervisor (por ejemplo, el modo de núcleo). En esencia, esta descentralización da como resultado dos niveles de infraestructura de programación que alivia una porción significativa de la carga de programación del modo de supervisor. Las operaciones básicas que son privilegiadas o comportan recursos de todo el sistema, tales como, por ejemplo, multiplexar un procesador físico a reservas y procesar interrupciones de temporizador, se proporcionan en el modo de supervisor (utilizando el programador / formato de objeto de ejecución del modo de supervisor). Por ejemplo, los procesadores virtuales de modo de supervisor pueden reservar recursos de procesador físicos en el modo de supervisor. Los dominios de programación de modo protegido se ejecutan en el modo protegido y multiplexan objetos de ejecución en procesadores virtuales. Por consiguiente, una cantidad significativa de operaciones de programación que, de lo contrario, se realizarían en el modo de supervisor, se realizan en su lugar en el modo protegido.

Además, el trabajo para una aplicación se puede realizar usando objetos de ejecución que difieren del modo de supervisor de formato de objeto de ejecución por defecto 103 y / o usando un programador que difiere del programador por defecto del modo de supervisor 130. Por ejemplo, un dominio de programación de modo protegido se puede configurar con un programador y un formato de objeto de ejecución que son más convenientes para el trabajo de una aplicación especificada.

Las formas de realización de la invención también incluyen usar el formato de objeto de ejecución del modo de supervisor y / o algoritmo de programación del modo de supervisor dentro de un dominio de programación protegido. Por ejemplo, el objeto de ejecución 131 y 132 puede ser del formato de objeto de ejecución (por ejemplo, subprocesos) que se usa en el modo de supervisor 103 y el algoritmo de directiva de programación 107 puede ser el que se usa en el modo de supervisor 103 (por ejemplo, BVT). Incluso en este entorno, la programación descentralizada sigue atenuando una porción de la carga de programación del modo de supervisor 103.

La figura 1C ilustra una porción de la arquitectura informática 100 a modo de ejemplo que muestra componentes adicionales de un procesador virtual. En esencia, el procesador virtual de modo protegido 108P simula el comportamiento del procesador físico 109. El procesador virtual de modo protegido 108 puede usar dos modelos de notificación, los paradigmas de interrupción y de puerto de terminación, para eventos asíncronos. La conmutación de contexto tiene lugar en el distribuidor 126 sin comportar llamada alguna al modo de supervisor 103. El distribuidor 126 puede conmutar el contenido a objetos de ejecución, por ejemplo, encima de una cola ejecutable.

En general, los manejadores de notificaciones son responsables de procesar las solicitudes de notificación que son generadas por el sistema operativo 101, tales como, por ejemplo, temporizadores y sincronización fuera del dominio de programación de modo protegido 104. El dominio de programación de modo protegido 104 puede establecer temporizadores o bien para el adelantamiento o bien para la carga de trabajo. El modo de supervisor 103 puede generar una interrupción virtual cuando expira el temporizador. El manejador de notificaciones de temporizador 123 procesa interrupciones virtuales que son resultado de temporizadores expirados. Los eventos de sincronización fuera del dominio de programación de modo protegido 104 también pueden generar interrupciones virtuales. El manejador de notificaciones de sincronización 124 puede procesar interrupciones virtuales que son resultado de los eventos de sincronización.

Las llamadas de interfaz binaria de aplicación (“ABI”, *Application Binary Interface*) sin Bloqueo pueden dar lugar a que un procesador virtual 108 detenga la espera para que se complete la operación. Por lo tanto, estos tipos de llamadas se pueden emitir en el modo de supervisor 103, generando una interrupción a la terminación. El manejador de notificaciones de devolución de llamada 125 puede procesar interrupciones virtuales que son resultado de llamadas de ABI sin bloqueo.

Los vectores de interrupción 122 almacenan índices o direcciones de memoria en una tabla de vectores de interrupción (contienen direcciones de memoria) para el manejador de notificaciones de temporizador 123, el manejador de notificaciones de sincronización 124 y el manejador de notificaciones de devolución de llamada 125. Por lo tanto, cuando se recibe una interrupción virtual, el controlador 121 puede hacer referencia a los vectores de interrupción 122 para ejecutar el manejador apropiado en respuesta a la interrupción virtual. El procesador virtual de modo de supervisor 108S puede comunicar interrupciones al procesador virtual de modo protegido 108P a través de la interfaz de interrupciones virtuales 136.

Los datos de terminación se pueden comunicar desde el procesador virtual de modo de supervisor 108S al dominio de programación de modo protegido 104 a través de la interfaz de puerto de terminación 138. Cuando se usa el paradigma de puertos de terminación, un manejador de notificaciones notifica al distribuidor 126 con una interrupción virtual. En respuesta, se detiene la ejecución del objeto de ejecución actual y el control se pasa al algoritmo de directiva de programación 107. Cuando un evento asíncrono está configurado para entregarse a la interfaz de puerto de terminación 138, con la terminación del evento, el procesador virtual de modo de supervisor 108S puede poner en cola el evento en su cola de notificación.

El algoritmo de directiva de programación 107 puede procesar los eventos completados pendientes en la cola cuando se presente la siguiente oportunidad. El adelantamiento puede no tener lugar y la ejecución del objeto de ejecución en ejecución continúa sin interrupción a partir de un evento completado. Durante la inicialización de un proceso, el algoritmo de directiva de programación 107 configura el mecanismo de notificación que se utilizará en los tipos diferentes de eventos.

Por lo tanto, el procesador virtual de modo de supervisor 108S puede llamar al K-VP para el controlador 121 o poner en cola eventos en el puerto de terminación cuando se completan los eventos asíncronos. En efecto, el controlador 121 se ejecuta en la pila del objeto de ejecución actual. Entonces, el algoritmo de directiva de programación 107 puede gestionar la pila y la conmutación de contexto después de que se haya iniciado la notificación. El algoritmo de directiva de programación 107 puede deshabilitar / habilitar interrupciones virtuales al tiempo que se procesan notificaciones o se ejecuta código de cambio de estado de objeto de ejecución. Se puede implementar un modelo similar a un error doble y triple para abordar las excepciones.

En general, un proceso de programación de reserva de procesadores físicos al menos dos formas para asignar procesadores virtuales. Un proceso puede reservar recursos de procesador físicos (por ejemplo, del procesador físico 109) y adjuntar un código auxiliar de programador de modo protegido que implementa el procesador virtual de modo de supervisor (por ejemplo, 108S). El código auxiliar de programador de modo protegido crea un objeto de ejecución basado en núcleo (por ejemplo, un subproceso) y expone el mismo al proceso como un procesador virtual (por ejemplo, 108). Además, el código auxiliar de programador de modo protegido genera interrupciones virtuales para notificar al controlador de procesador virtual de modo protegido (por ejemplo, 121). El procesador virtual consume la totalidad de los recursos que se asignan a la reserva. Por lo tanto, usando un patrón de uso de reserva, un único procesador virtual es expuesto por el modo de supervisor.

En algunas formas de realización, no se utiliza una reserva rápida y, en su lugar, se solicita un procesador virtual a partir de una reserva por defecto. En estas formas de realización, al objeto de ejecución de modo de supervisor (por ejemplo, un subproceso de núcleo) que representa el programador por defecto de modo de supervisor se le asignan recursos en función de la directiva. Por ejemplo, si se aplica un algoritmo de operación por turnos, el procesador virtual de modo de supervisor puede compartir por igual en la reserva por defecto con la totalidad de los otros objetos de ejecución de modo de supervisor y procesadores virtuales de modo de supervisor que operan en la reserva por defecto. Las directivas de programador de modo de supervisor se pueden extender para incorporar una funcionalidad de código auxiliar de programador de modo protegido, tal como, por ejemplo, exponer procesadores virtuales a procesos y generar interrupciones virtuales.

Un dominio de programación de modo protegido puede encapsular múltiples procesadores virtuales. De acuerdo con los requisitos de la aplicación, los procesadores virtuales se pueden asignar en procesadores físicos diferentes o el mismo procesador. Procesadores virtuales diferentes también se pueden asignar a operaciones de reserva diferentes. En algunas formas de realización, múltiples procesadores virtuales se asignan usando una reserva de conjunto. La reserva además de los requisitos de recursos especifica un factor de simultaneidad. El modo de supervisor reserva recursos para múltiples procesadores físicos y asigna los mismos de tal modo que los procesadores virtuales se ejecutan de forma simultánea en todos los procesos. No obstante, también es posible asignar una mezcla de procesadores virtuales a partir de la reserva por defecto y reservas rápidas.

La figura 1D ilustra una porción de la arquitectura informática 100 a modo de ejemplo que muestra múltiples procesadores virtuales en un dominio de programación de modo protegido. Tal como se muestra, el dominio de

programación de modo protegido 104 incluye los procesadores virtuales de modo protegido 108P y 128P. Los procesadores virtuales de modo protegido 108P y 128P utilizan los algoritmos de directiva de programación 107 y 127 correspondientes, de forma respectiva. Cada uno de los algoritmos de directiva de programación 107 y 127 conserva el estado de objeto de ejecución en el almacén de objetos de ejecución 105 y su propio estado de directiva correspondiente en el estado de directiva 116.

La semántica del dominio de programación de modo protegido 104 se puede usar para organizar el almacén de objetos de ejecución 105 y el almacén de directivas 116 y compartir información entre los algoritmos de directiva de programación 107 y 127. Si, por ejemplo, una directiva soporta un equilibrado dinámico de cargas entre los procesadores virtuales, se puede usar un almacén común. No obstante, si una directiva soporta la afinación a objetos de ejecución, se pueden usar almacenes separados. En efecto, el mismo programador se podría ejecutar en todos los procesadores virtuales o programadores especializados se podrían ejecutar en procesadores virtuales diferentes dependiendo de los requisitos de la carga de trabajo.

La figura 1E ilustra una porción de la arquitectura informática 100 a modo de ejemplo que muestra componentes para sincronizar entre los dominios de programación de modo protegido. Tal como se muestra, el proceso 181 y el proceso 182 se están ejecutando en el modo protegido 102. El proceso 181 incluye los dominios de programación de modo protegido 104 y 144, los procesadores virtuales de modo protegido 108P y 148P y la capa de sincronización 141. El proceso 182 incluye los dominios de programación de modo protegido 154 y los procesadores virtuales de modo protegido 158P. Los procesadores virtuales de modo de supervisor 108S, 148S y 158S, se ejecutan en el modo de supervisor 103 y se corresponden con los procesadores virtuales 108P, 148P, y 158P, de forma respectiva. El modo de supervisor 103 también incluye la capa de sincronización 142.

Las operaciones de sincronización dentro de un dominio de programación de modo protegido son manejadas por el programador de modo protegido sin implicación alguna del modo de supervisor. Por ejemplo, la sincronización dentro del dominio de programación de modo protegido 104, 144 o 154 se puede manejar sin referencia a capas de sincronización externas. En algunas formas de realización, la sincronización de objetos de ejecución entre programadores de modo protegido también se realiza sin implicación alguna del modo de supervisor. Por ejemplo, la sincronización entre los dominios de programación de modo protegido 104 y 144 se puede manejar en la capa de sincronización 141 dentro del proceso 182.

En otras formas de realización, también se realiza una sincronización de objetos de ejecución entre procesos. Por ejemplo, la sincronización entre uno u otro de los dominios de programación de modo protegido 104 y 144 y el dominio de programación de modo protegido 154 se puede manejar en la capa de sincronización 142 dentro del modo de supervisor 103.

Los eventos de canal se pueden usar para facilitar la sincronización. Un evento de canal es un evento en el que participan dos objetos de ejecución a partir de dominios de programación de modo protegido específicos. Por ejemplo, en un canal entre el proceso 181 y el proceso 182, un objeto de ejecución a partir del dominio de programación de modo protegido 144 y un objeto de ejecución a partir del dominio de programación de modo protegido 154 pueden participar en un evento de canal que soporta una comunicación inter-proceso. Un evento de canal inter-proceso se puede implementar como la capa de sincronización 142 en el modo de supervisor 103. Cuando se indica un evento de canal, el modo de supervisor 103 notifica al procesador virtual de modo de supervisor, lo que, a su vez, genera una interrupción virtual para notificar el evento al procesador virtual de modo protegido (por ejemplo, 148 o 158).

De forma similar, la misma operación podría tener lugar entre los dominios de programación de modo protegido dentro del mismo proceso. Por ejemplo, los dominios de programación de modo protegido 104 y 144 pueden requerir una operación de sincronización. Un evento de canal se puede usar para compensar formatos de objeto de ejecución potencialmente diferentes que se usan entre los dominios de programación de modo protegido 104 y 144. Un evento de canal intra-proceso se puede implementar como la capa de sincronización 141 en el proceso 181. La capa de sincronización 141 puede crear una interrupción virtual en los procesadores virtuales (por ejemplo, 108 o 148) que se va a notificar para un evento nuevo. Los programadores de modo protegido pueden aplicar el evento de canal en el contexto de su propio modelo de sincronización.

La infraestructura de programación de modo de supervisor se puede implementar de forma flexible de una diversidad de formas diferentes para proporcionar unas características apropiadas al modo protegido para crear dominios de programación de modo protegido. Las figuras 2A y 2B ilustran las infraestructuras de modo de supervisor 200 y 250 a modo de ejemplo que proporcionan características a los dominios de programación de modo protegido.

Tal como se muestra, la infraestructura de modo de supervisor 200 incluye el distribuidor 201, la directiva de programación por defecto 202, los procesadores virtuales de modo de supervisor 203, la directiva de programación de modo de supervisor 204, la sincronización 205, el almacén de objetos de ejecución 206 y la sincronización de eventos de canal 207. Cada uno de los componentes mostrados está operando en el modo de supervisor (por ejemplo, de núcleo) 231. El modo de supervisor 231 puede usar un objeto de ejecución por defecto (por ejemplo, subprocesos) para realizar operaciones.

La infraestructura de modo de supervisor 200 puede usar una reserva dedicada para procesar objetos de ejecución de modo de supervisor (por ejemplo, subprocesos de núcleo). Las operaciones del almacén de objetos de ejecución de modo de supervisor 206 y de la sincronización 205 están asociadas con la directiva de programación de modo de supervisor 204 rápida. Por otro lado, la directiva de programación por defecto 202 se usa para asignar los procesadores virtuales 203 (por ejemplo, realizar reservas de recursos) y multiplexar. Por lo tanto, el distribuidor 201 usa la directiva de programación por defecto 202 para asignar los procesadores virtuales 203 y multiplexar entre los mismos y la directiva de programación de modo de supervisor 204 para manejar la comunicación con el almacén de objetos de ejecución de modo de supervisor 206 y la sincronización 205. Dentro de la infraestructura de modo de supervisor 200, el modo de supervisor 231 y la directiva de programación por defecto 202 se pueden diseñar de forma específica para las características de las cargas de trabajo respectivas.

Tal como se muestra, las infraestructuras de modo de supervisor 250 incluyen el distribuidor 201, la directiva de programación por defecto 202, los procesadores virtuales de modo de supervisor 203, la sincronización 205, el almacén de objetos de ejecución 206 y la sincronización de eventos de canal 207. Cada uno de los componentes mostrados está operando en el modo de supervisor (por ejemplo, de núcleo) 231. La infraestructura de modo de supervisor 250 usa la directiva de programación por defecto 202 para manejar tanto objetos de ejecución de modo de supervisor como procesadores virtuales que se asignan a la reserva por defecto.

La figura 3 ilustra una arquitectura de programación de modo de usuario 300 a modo de ejemplo. La arquitectura de programación de modo de usuario 300 muestra un ejemplo de cómo se pueden combinar diversos componentes en la arquitectura informática 100 para proporcionar una programación de modo de usuario para objetos de ejecución. Tal como se muestra, el programador de modo de usuario 304 incluye el almacén de objetos de ejecución 305, el estado de directiva 316, la contabilización 309, el algoritmo de directiva 307 y la sincronización (dominio) 308. El programador de modo de usuario 304 puede asignar recursos de procesador al objeto de ejecución 181 u otros objetos de ejecución. La sincronización (dominio) 308 puede coordinar las transiciones de estado entre objetos de ejecución dentro del programador de modo de usuario 304.

El procesador virtual de usuario 321U está configurado para procesar notificaciones de modo de núcleo a partir del modo de núcleo 303 y distribuir objetos de ejecución dentro del programador de modo de usuario 304. El procesador virtual de usuario 321U puede simular el comportamiento del procesador físico 109. El procesador virtual de usuario 321U puede usar uno cualquiera de los dos modelos de notificación, los paradigmas de interrupción y de puerto de terminación, para eventos asíncronos. La conmutación de contexto tiene lugar en el distribuidor de modo de usuario 311 sin comportar llamada alguna al modo de supervisor 103. El distribuidor 311 puede conmutar el contenido a objetos de ejecución (por ejemplo, tareas pendientes) encima de una cola ejecutable.

En general, los manejadores de notificaciones 312 son responsables de procesar las solicitudes de notificación que son generadas por un sistema operativo, tales como, por ejemplo, temporizadores y sincronización fuera del programador de modo de usuario 304. El programador de modo de usuario 304 puede establecer temporizadores o bien para el adelantamiento o bien para la carga de trabajo. El modo de núcleo 303 puede generar una interrupción virtual cuando expira el temporizador. Los eventos de sincronización fuera del programador de modo de usuario 304 también pueden generar interrupciones virtuales. Las llamadas de interfaz binaria de aplicación (“ABI”, *Application Binary Interface*) sin Bloqueo se pueden emitir en el modo de núcleo 303, generando una interrupción a la terminación. Los manejadores de notificaciones 312 pueden manejar cualquiera de estos tipos de interrupciones.

Los vectores de interrupción 314 almacenan índices o direcciones de memoria en una tabla de vectores de interrupción (contienen direcciones de memoria) para los manejadores de notificaciones 312. Por lo tanto, cuando se recibe una interrupción virtual, el controlador de procesador virtual 313 puede hacer referencia a los vectores de interrupción 314 para ejecutar el manejador apropiado en respuesta a la interrupción virtual.

El distribuidor de núcleo 323 puede distribuir entre el procesador virtual de núcleo 321K y los subprocesos de núcleo de acuerdo con una directiva de programación por defecto o una combinación de una directiva de programación por defecto y una directiva de programación de núcleo 324 (por ejemplo, como en las arquitecturas de modo de supervisor 200 y 250).

La sincronización de canal (intra-proceso) 318 puede sincronizar entre el programador de modo de usuario 304 y otros programadores de modo de usuario en el mismo proceso con el programador de modo de usuario 304. La sincronización de canal (inter-proceso) 328 puede sincronizar entre el programador de modo de usuario 304 y programadores de modo de usuario en los procesos diferentes.

La contabilización 309 y la contabilidad de sistema 322 interaccionan para gestionar funciones de contabilización, tales como, por ejemplo, el uso de recursos de procesador físico.

Por consiguiente, la programación del modo protegido (por ejemplo, de modo de usuario) puede facilitar el desarrollo de marcos de trabajo de programación que reflejan mejor los requisitos de las cargas de trabajo a través del uso de abstracciones de ejecución específicas de la carga de trabajo. Además, la capacidad de definir unas directivas de programación optimizados para las características de los recursos de soporte físico disponibles y los requisitos de carga de trabajo presenta el potencial de unas características mejores de cambio de escala de sistemas. Además,

tal como se ha descrito previamente, la programación de modo protegido descentraliza la responsabilidad de la programación mediante el movimiento de unas porciones significativas de funcionalidad de programación desde el modo de supervisor (por ejemplo, el modo de núcleo) a una aplicación.

La presente invención se puede materializar en otras formas específicas sin apartarse de sus características esenciales. Las formas de realización que se describen se han de considerar en todo aspecto solo como ilustrativas y no restrictivas. El ámbito de la invención se indica, por lo tanto, por las reivindicaciones adjuntas en lugar de por la descripción anterior. Todos los cambios que entren dentro del significado y el rango de equivalencia de las reivindicaciones se han de englobar dentro de su ámbito.

REIVINDICACIONES

1. En un sistema informático, incluyendo el sistema informático un procesador físico (109) y un sistema operativo (101) que tiene un modo protegido (102) y un modo de supervisor (103), en el que el modo protegido (102) limita el acceso de los programas a recursos de procesador físico de los programas en ejecución en el modo protegido (102), teniendo el modo de supervisor (103) un acceso ilimitado a los recursos de procesador físico, controlando una o más funciones el acceso al modo de supervisor (103) para permitir que un programa en el modo protegido (102) realice una transición al modo de supervisor (103) para acceder a recursos de procesador físico, en el que los programas que se ejecutan dentro del sistema operativo (101) pueden realizar una transición desde el modo protegido (102) al modo de supervisor (103) y desde el modo de supervisor (103) al modo protegido (102) dependiendo de instrucciones de programa, teniendo el modo de supervisor (103) un formato de objeto de ejecución de modo de supervisor por defecto que son subprocesos que consume recursos de procesador físico de una forma compartida en el tiempo de acuerdo con una directiva de programación de modo de supervisor por defecto, en el que un objeto de ejecución es una instancia de una abstracción de carga de trabajo que consume recursos y se comparte en el tiempo en el procesador físico, un procedimiento para programar el consumo de recursos de procesador en el modo protegido, comprendiendo el procedimiento:

un acto de asignar (112) recursos de procesador físico a un primer objeto de ejecución (131) de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido (104), recursos de procesador físico que se asignan de acuerdo con una directiva de programación de modo protegido (107) que difiere de la directiva de programación de modo de supervisor por defecto, en el que la directiva de programación de modo protegido es para multiplexar objetos de ejecución del dominio de programación de modo protegido en un procesador virtual, asignando el procesador virtual al menos una porción del procesador físico (109) para su uso por objetos de ejecución del dominio de programación de modo protegido (104);
un acto de utilizar los recursos del procesador físico (112) asignado para completar parcialmente el trabajo que se indica en el primer objeto de ejecución (131);
un acto de determinar, de acuerdo con la directiva de programación de modo protegido (107), que la asignación de recursos de procesador físico (112) va a realizar una transición a un segundo objeto de ejecución (132) dentro del dominio de programación de modo protegido antes de completar completamente el trabajo que se indica en el primer objeto de ejecución (131);
un acto de conmutar el contexto desde el primer objeto de ejecución al segundo objeto de ejecución dentro del modo protegido y sin realizar una transición al modo de supervisor, que incluye:

un acto de conservar el estado del primer objeto de ejecución (133) en un almacén de objetos de ejecución (105), el almacén de objetos de ejecución (105) residente dentro del dominio de programación de modo protegido (104); y
un acto de cargar el estado del segundo objeto de ejecución (134) a partir del almacén de objetos de ejecución (105);

un acto de asignar recursos de procesador físico (112) al segundo objeto de ejecución (132) de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido (104) posteriormente a la conmutación del contexto al segundo objeto de ejecución (132); y
un acto de utilizar los recursos de procesador físicos asignados (112) para completar parcialmente el trabajo que se indica en el segundo objeto de ejecución;
en el que el modo protegido comprende modo de usuario y el modo de supervisor comprende modo de núcleo, y
en el que el modo de núcleo usa subprocesos y en el que el dominio de programación de modo protegido es un dominio de programación de modo de usuario que usa un formato de objeto de ejecución que no sean subprocesos.

2. El procedimiento según la reivindicación 1, en el que el acto de asignar recursos de procesador físico a un primer objeto de ejecución de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido comprende un acto de asignar recursos de procesador físico a un objeto de ejecución en modo de usuario del sistema operativo; y / o en el que asignar recursos de procesador físico a un primer objeto de ejecución comprende asignar recursos de procesador físico que son reservados por un procesador virtual de modo de núcleo.

3. El procedimiento según las reivindicaciones 1 a 2, que comprende adicionalmente:

un acto de crear el dominio de programación de modo protegido (104) que opera en el modo protegido (102) del sistema operativo (101), incluyendo la creación del dominio de programación de modo protegido (104):

crear el procesador virtual (108);
cargar la directiva de programación de modo protegido (107); y
crear el almacén de objetos de ejecución (105) que está configurado para almacenar el estado para objetos de ejecución del dominio de programación de modo protegido (104) para facilitar la conmutación del contexto entre diferentes objetos de ejecución dentro del dominio de programación de modo protegido.

4. Un sistema informático, comprendiendo el sistema informático:

uno o más procesadores físicos (109);
memoria de sistema;
uno o más medios de almacenamiento físicos que tienen, almacenados en los mismos, instrucciones ejecutables por ordenador que representan un sistema operativo (101) que tiene un modo protegido (102) y un modo de supervisor (103) que, cuando se ejecutan en uno de los procesadores, dan lugar a que el sistema informático realice lo siguiente:

asignar recursos de procesador físico a un primer objeto de ejecución (131) de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido (104), recursos de procesador físico que se asignan de acuerdo con una directiva de programación de modo protegido (107) que difiere de una directiva de programación de modo de supervisor por defecto, en el que un objeto de ejecución es una instancia de una abstracción de carga de trabajo que consume recursos y se comparte en el tiempo en el procesador físico, en el que el modo de supervisor (103) tiene un formato de objeto de ejecución de modo de supervisor por defecto que son subprocesos que consume recursos de procesador físico de una forma compartida en el tiempo de acuerdo con la directiva de programación de modo de supervisor por defecto, en el que la directiva de programación de modo protegido es para multiplexar objetos de ejecución del dominio de programación de modo protegido en un procesador virtual, asignando el procesador virtual al menos una porción del procesador físico (109) para su uso por objetos de ejecución del dominio de programación de modo protegido (104);
utilizar los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el primer objeto de ejecución;
determinar, de acuerdo con la directiva de programación de modo protegido, que la asignación de recursos de procesador físico va a realizar una transición a un segundo objeto de ejecución dentro del dominio de programación de modo protegido antes de completar completamente el trabajo que se indica en el primer objeto de ejecución;
conmutar el contexto desde el primer objeto de ejecución al segundo objeto de ejecución dentro del modo protegido y sin realizar una transición al modo de supervisor, que incluye:

un acto de conservar el estado del primer objeto de ejecución en un almacén de objetos de ejecución, el almacén de objetos de ejecución residente dentro del dominio de programación de modo protegido; y
un acto de cargar el estado del segundo objeto de ejecución a partir del almacén de objetos de ejecución;

asignar recursos de procesador físico al segundo objeto de ejecución de un dominio de programación de modo protegido posteriormente a la conmutación del contexto al segundo objeto de ejecución; y
utilizar los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el segundo objeto de ejecución,
en el que el modo protegido comprende modo de usuario y el modo de supervisor comprende modo de núcleo, y
en el que el modo de núcleo usa subprocesos y en el que el dominio de programación de modo protegido es un dominio de programación de modo de usuario que usa un formato de objeto de ejecución que no sean subprocesos.

5. El sistema según la reivindicación 4, que comprende adicionalmente:

instrucciones ejecutables por ordenador que, cuando se ejecutan, dan lugar a que el sistema informático: cree el dominio de programación de modo protegido (104) que opera en el modo protegido (102) del sistema operativo (101), incluyendo la creación del dominio de programación de modo protegido (104):

crear el procesador virtual (108);
cargar la directiva de programación de modo protegido (107); y
crear el almacén de objetos de ejecución (105) que está configurado para almacenar el estado para objetos de ejecución del dominio de programación de modo protegido (104) para facilitar la conmutación del contexto entre diferentes objetos de ejecución dentro del dominio de programación de modo protegido.

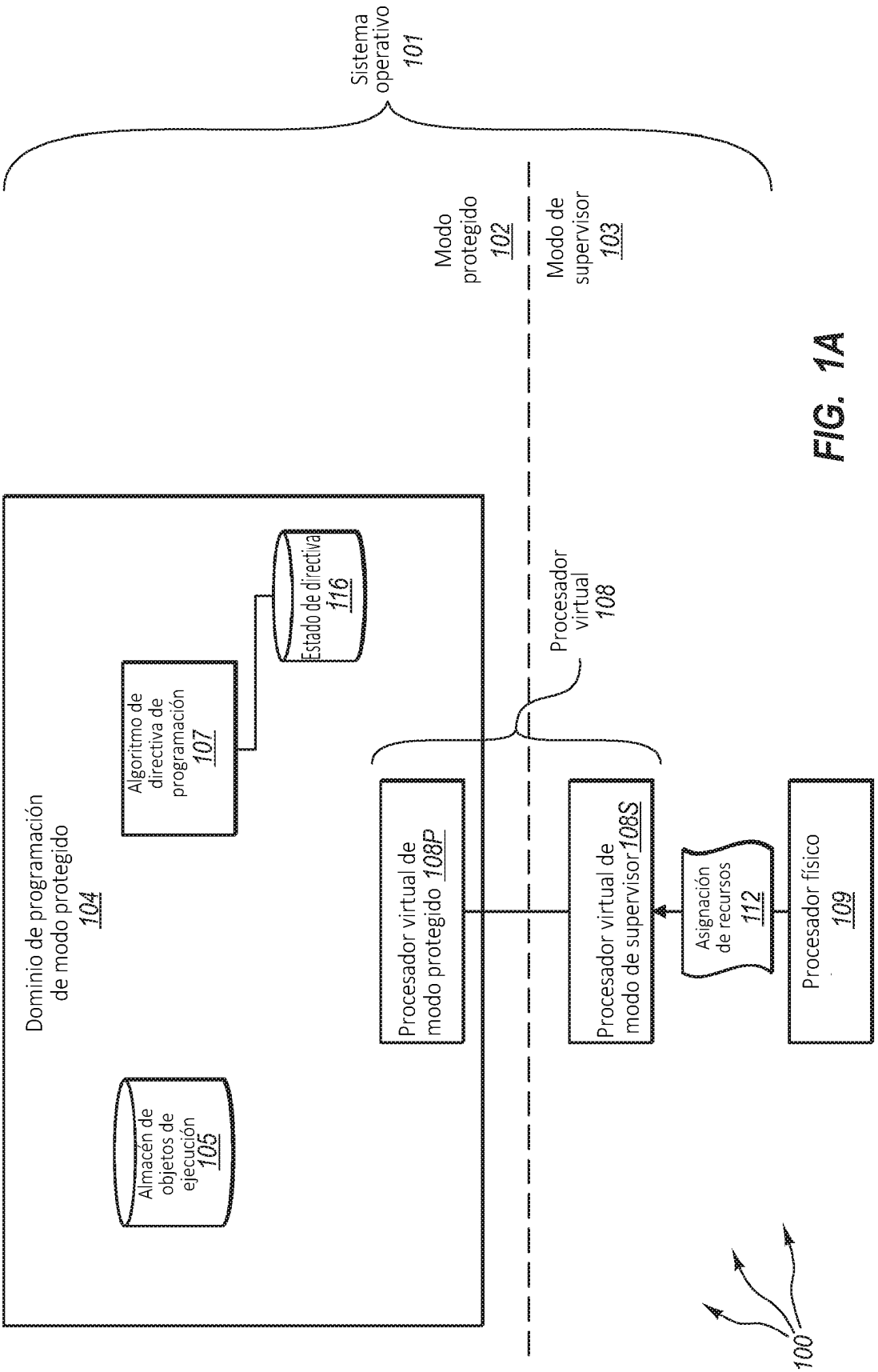
Sistema operativo 101
Modo protegido 102
Modo de supervisor 103
Dominio de programación de modo protegido 104
Almacén de objetos de ejecución 105
Algoritmo de directiva de programación 107
Estado de directiva 116
Procesador virtual 108
Procesador virtual de modo protegido 108P
Procesador virtual de modo de supervisor 108S
Asignación de recursos 112
Procesador físico 109
100

FIG. 1A

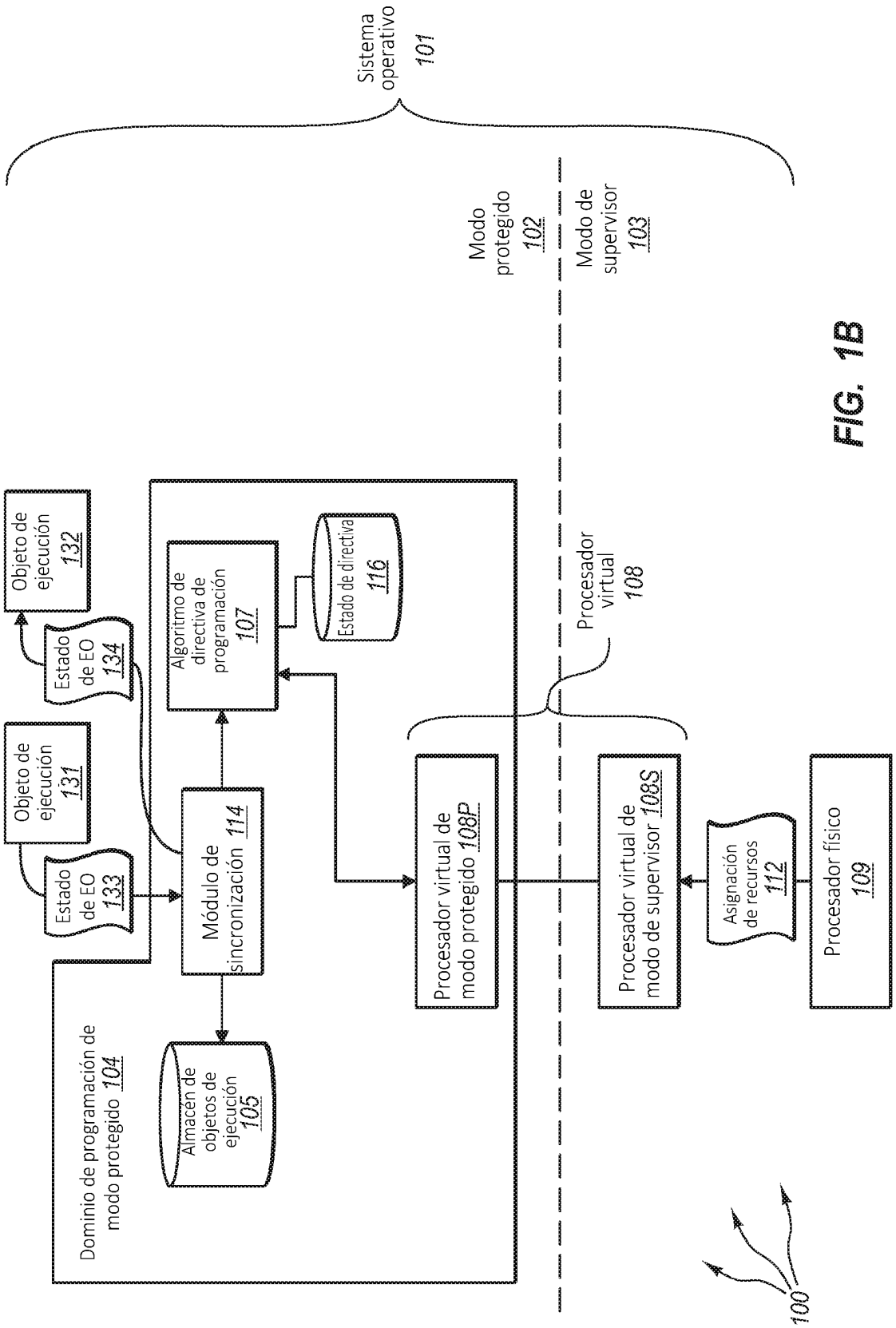
Sistema operativo 101
Modo protegido 102
Modo de supervisor 103
Procesador virtual 108
Dominio de programación de modo protegido 104
Objeto de ejecución 132
Estado de EO 134
Objeto de ejecución 131
Estado de EO 133
Módulo de sincronización 114
Algoritmo de directiva de programación 107
Estado de directiva 116
Almacén de objetos de ejecución 105
Procesador virtual de modo protegido 108P
Procesador virtual de modo de supervisor 108S
Asignación de recursos 112
Procesador físico 109
100

FIG. 1B

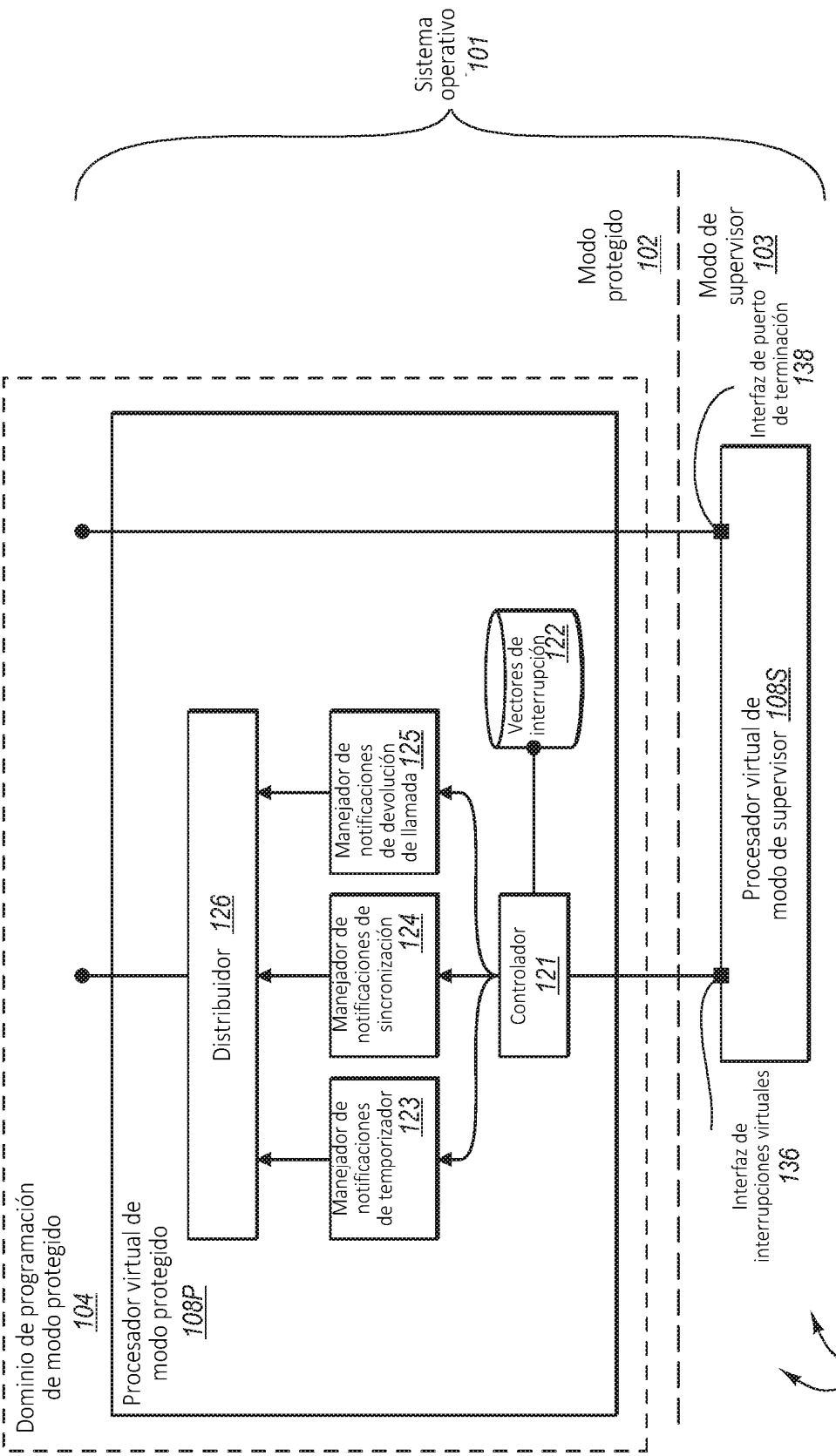
Sistema operativo 101
Modo protegido 102
Modo de supervisor 103
Interfaz de puerto de terminación 138
Dominio de programación de modo protegido 104
Procesador virtual de modo protegido 108P
Distribuidor 126
Manejador de notificaciones de temporizador 123
Manejador de notificaciones de sincronización 124
Manejador de notificaciones de devolución de llamada 125
Controlador 121
Vectores de interrupción 122
Procesador virtual de modo de supervisor 108S
Interfaz de interrupciones virtuales 136
100

FIG. 1C

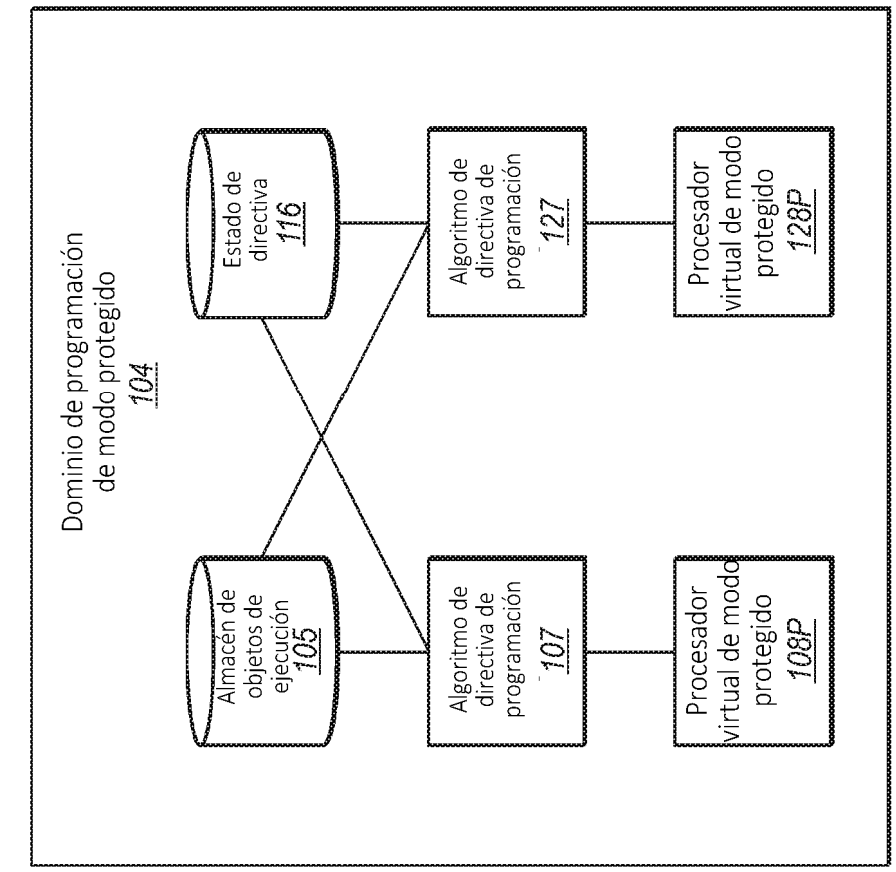
Dominio de programación de modo protegido
104
Estado de directiva
116
Algoritmo de directiva de programación
127
Procesador virtual de modo protegido
128P
Almacén de objetos de ejecución
105
Algoritmo de directiva de programación
107
Procesador virtual de modo protegido
108P
100

FIG. 1D

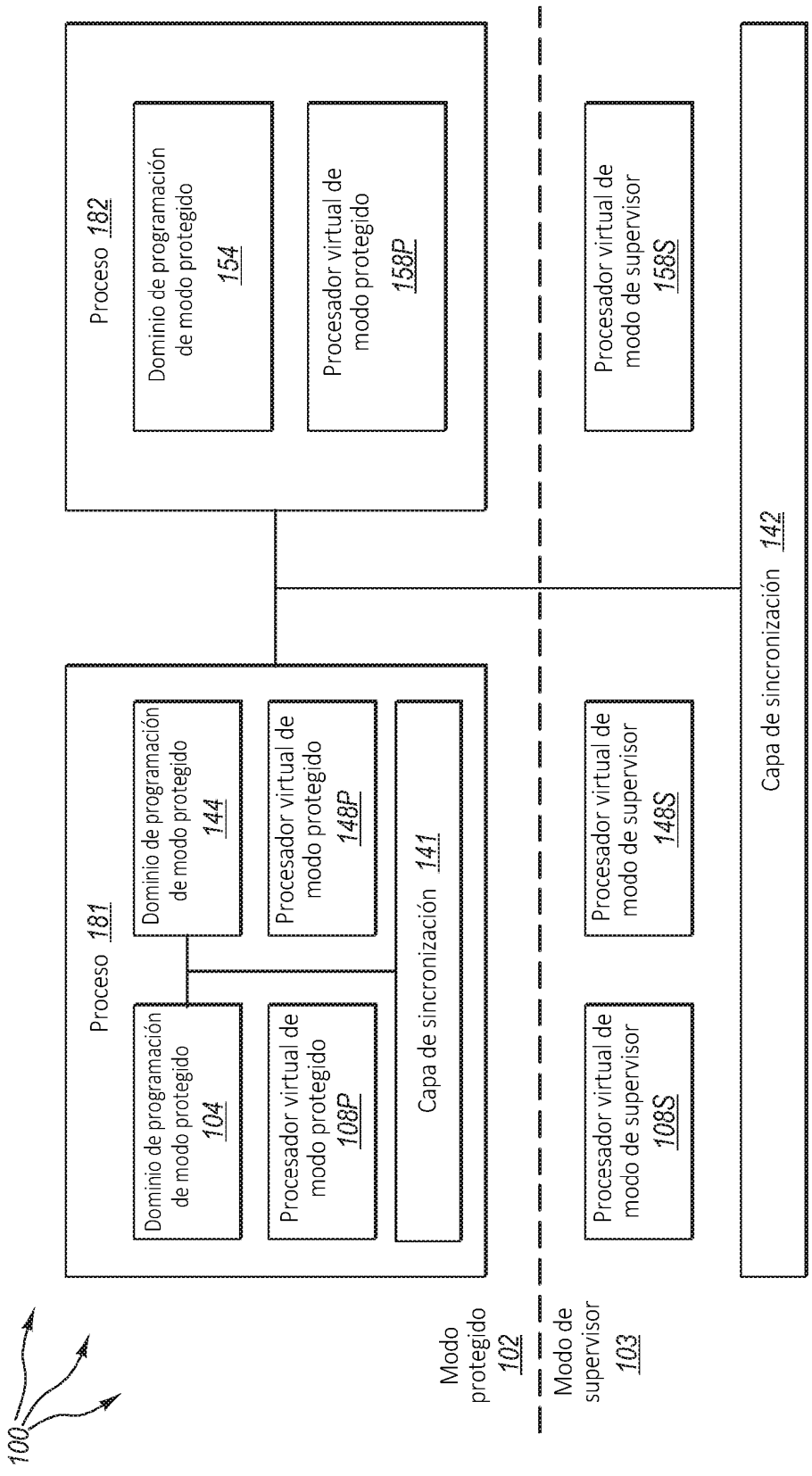
100
Modo protegido 102
Modo de supervisor 103
Proceso 181
Dominio de programación de modo protegido 104
Procesador virtual de modo protegido 108P
Capa de sincronización 141
Dominio de programación de modo protegido 144
Procesador virtual de modo protegido 148P
Procesador virtual de modo de supervisor 148S
Procesador virtual de modo de supervisor 108S
Proceso 182
Dominio de programación de modo protegido 154
Procesador virtual de modo protegido 158P
Procesador virtual de modo de supervisor 158S
Capa de sincronización 142

FIG. 1E

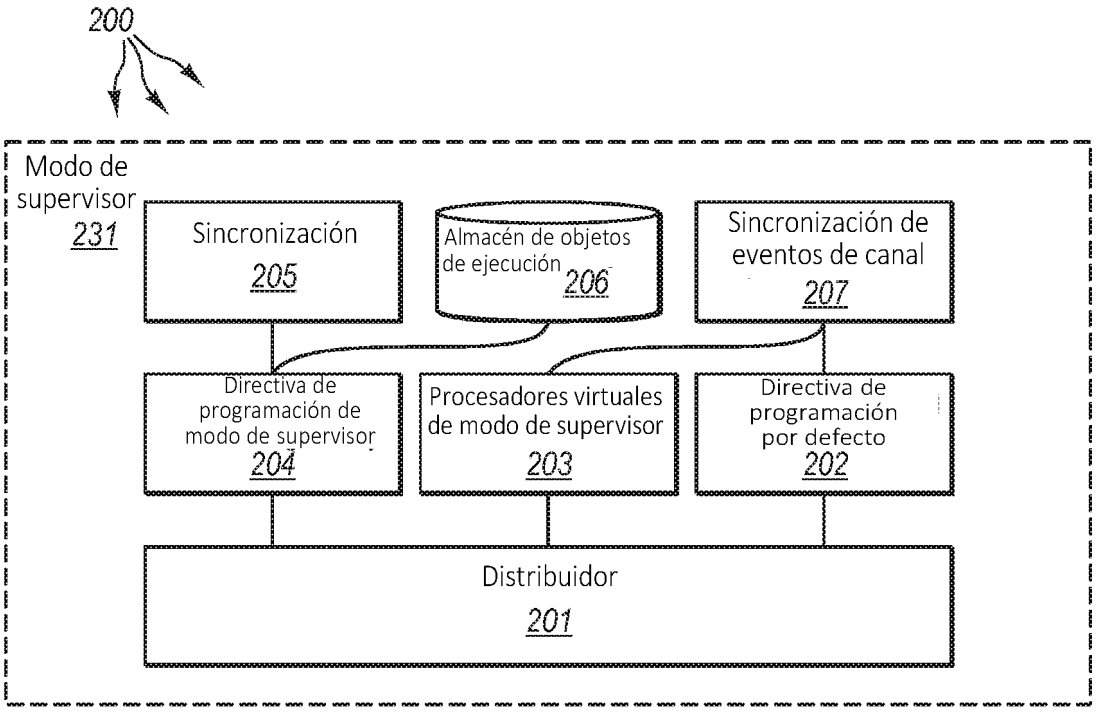
200
Modo de supervisor 231
Sincronización 205
Almacén de objetos de ejecución 206
Sincronización de eventos de canal 207
Directiva de programación de modo de supervisor 204
Procesadores virtuales de modo de supervisor 203
Directiva de programación por defecto 202
Distribuidor 201

FIG. 2A

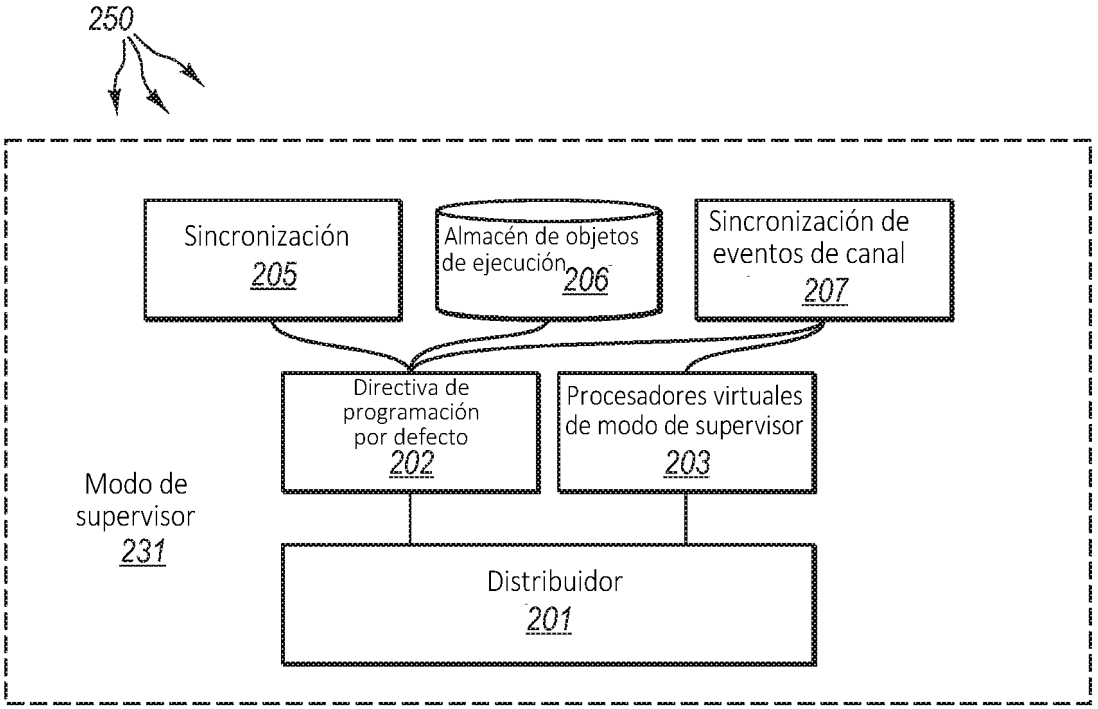
250
Sincronización 205
Almacén de objetos de ejecución 206
Sincronización de eventos de canal 207
Directiva de programación por defecto 202
Procesadores virtuales de modo de supervisor 203
Modo de supervisor 231
Distribuidor 201

FIG. 2B

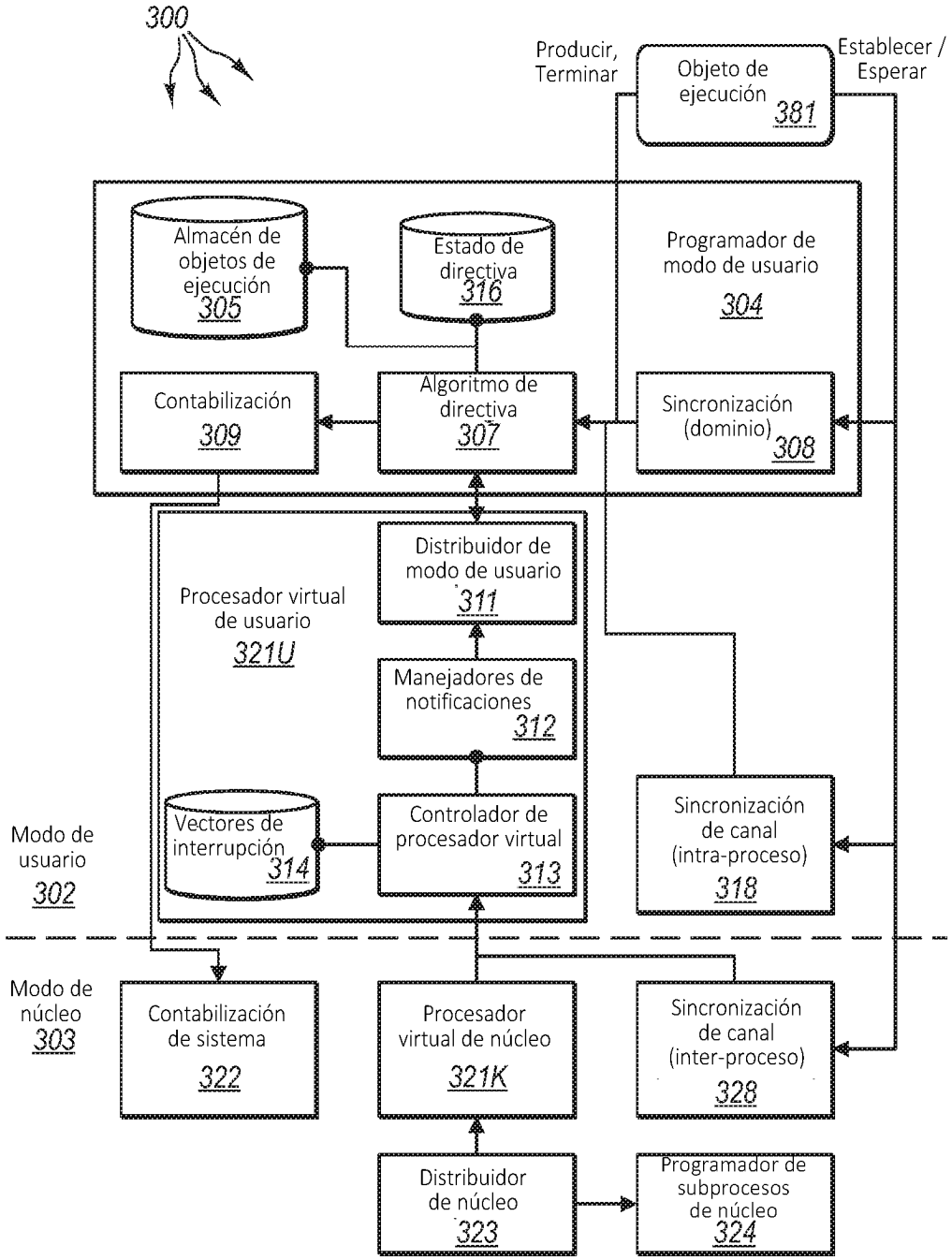
300
Producir, Terminar
Objeto de ejecución 381
Establecer / Esperar
Almacén de objetos de ejecución 305
Estado de directiva 316
Programador de modo de usuario 304
Contabilización 309
Algoritmo de directiva 307
Sincronización (dominio) 308
Distribuidor de modo de usuario 311
Procesador virtual de usuario 321U
Manejadores de notificaciones 312
Vectores de interrupción 314
Controlador de procesador virtual 313
Sincronización de canal (intra-proceso) 318
Modo de usuario 302
Modo de núcleo 303
Contabilización de sistema 322
Procesador virtual de núcleo 321K
Sincronización de canal (inter-proceso) 328
Distribuidor de núcleo 323
Programador de subprocesos de núcleo 324

FIG. 3

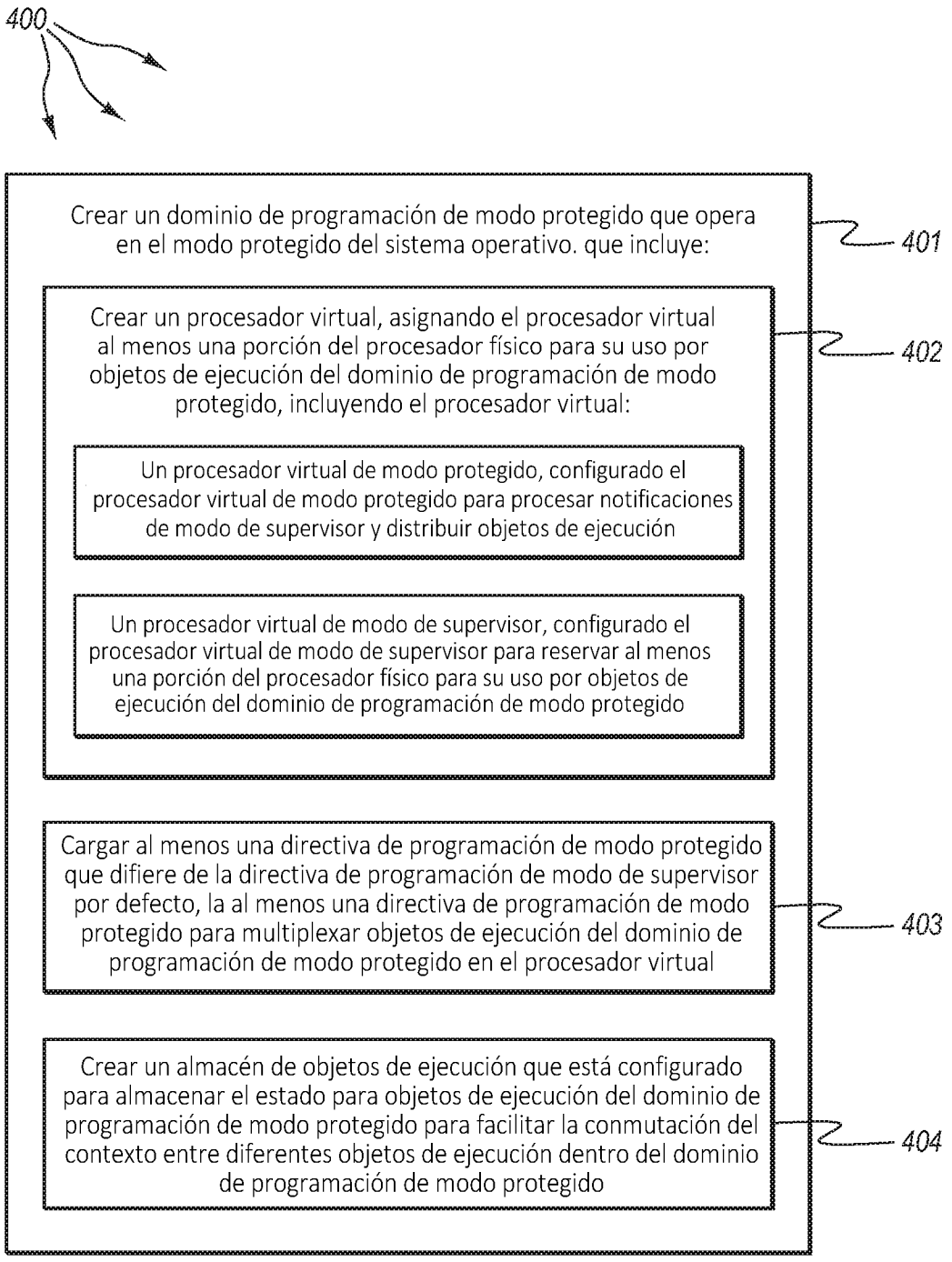
400
Crear un dominio de programación de modo protegido que opera en el modo protegido del sistema operativo. que incluye:
401
Crear un procesador virtual, asignando el procesador virtual al menos una porción del procesador físico para su uso por objetos de ejecución del dominio de programación de modo protegido, incluyendo el procesador virtual:
402
Un procesador virtual de modo protegido, configurado el procesador virtual de modo protegido para procesar notificaciones de modo de supervisor y distribuir objetos de ejecución
Un procesador virtual de modo de supervisor, configurado el procesador virtual de modo de supervisor para reservar al menos una porción del procesador físico para su uso por objetos de ejecución del dominio de programación de modo protegido
Cargar al menos una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto, la al menos una directiva de programación de modo protegido para multiplexar objetos de ejecución del dominio de programación de modo protegido en el procesador virtual
403
Crear un almacén de objetos de ejecución que está configurado para almacenar el estado para objetos de ejecución del dominio de programación de modo protegido para facilitar la conmutación del contexto entre diferentes objetos de ejecución dentro del dominio de programación de modo protegido
404

FIG. 4

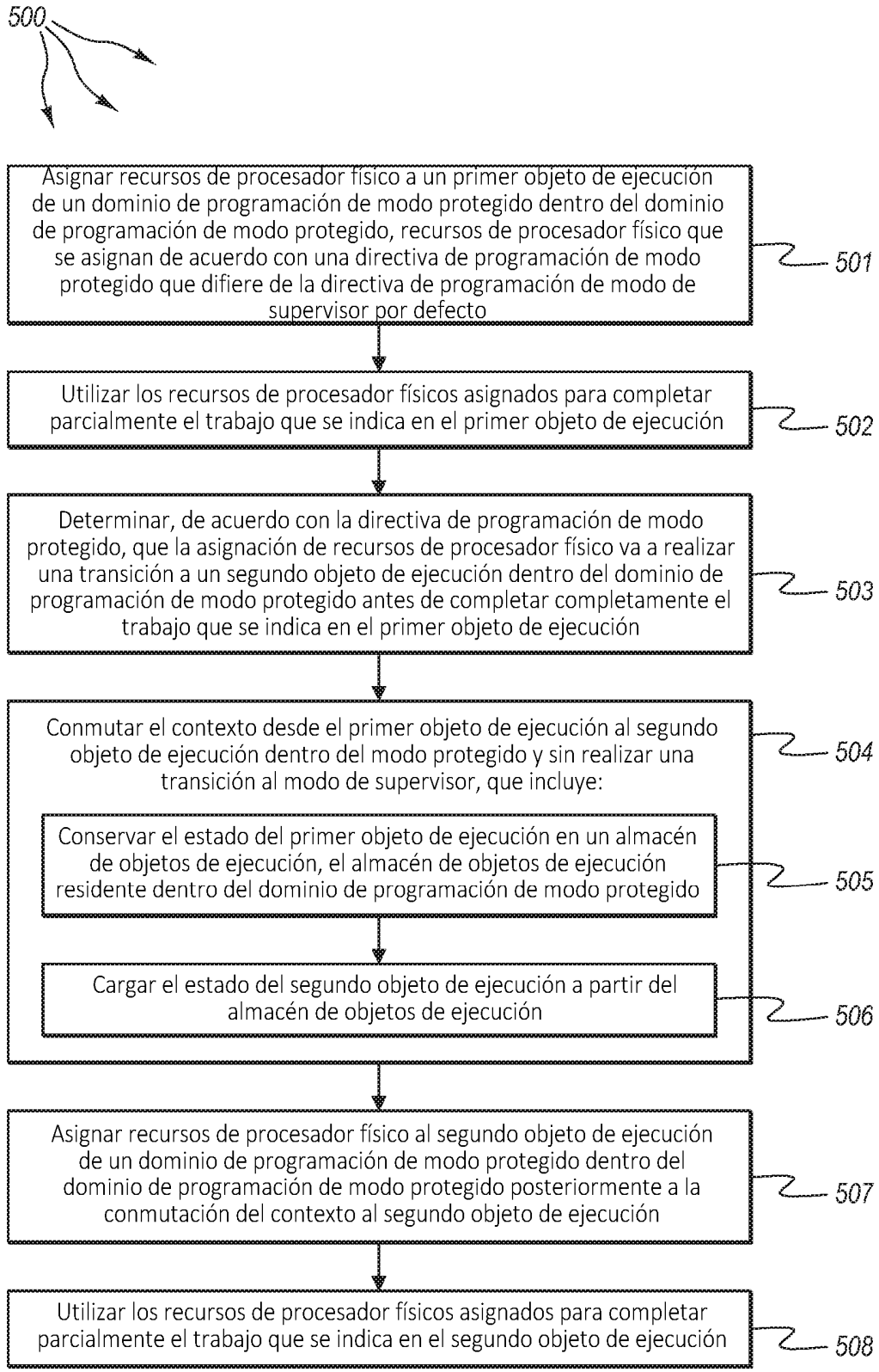
500
Asignar recursos de procesador físico a un primer objeto de ejecución de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido, recursos de procesador físico que se asignan de acuerdo con una directiva de programación de modo protegido que difiere de la directiva de programación de modo de supervisor por defecto
501
Utilizar los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el primer objeto de ejecución
502
Determinar, de acuerdo con la directiva de programación de modo protegido, que la asignación de recursos de procesador físico va a realizar una transición a un segundo objeto de ejecución dentro del dominio de programación de modo protegido antes de completar completamente el trabajo que se indica en el primer objeto de ejecución
503
Conmutar el contexto desde el primer objeto de ejecución al segundo objeto de ejecución dentro del modo protegido y sin realizar una transición al modo de supervisor, que incluye:
504
Conservar el estado del primer objeto de ejecución en un almacén de objetos de ejecución, el almacén de objetos de ejecución residente dentro del dominio de programación de modo protegido
505
Cargar el estado del segundo objeto de ejecución a partir del almacén de objetos de ejecución
506
Asignar recursos de procesador físico al segundo objeto de ejecución de un dominio de programación de modo protegido dentro del dominio de programación de modo protegido posteriormente a la conmutación del contexto al segundo objeto de ejecución
507
Utilizar los recursos de procesador físicos asignados para completar parcialmente el trabajo que se indica en el segundo objeto de ejecución
508

FIG. 5