

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 407 632**

51 Int. Cl.:

G06F 12/02 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **03.08.2009 E 09009987 (0)**

97 Fecha y número de publicación de la concesión europea: **20.03.2013 EP 2151761**

54 Título: **Gestión de memoria en un soporte de datos portátil**

30 Prioridad:

05.08.2008 DE 102008036479

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

13.06.2013

73 Titular/es:

**GIESECKE & DEVRIENT GMBH (100.0%)
PRINZREGENTENSTRASSE 159
81677 MÜNCHEN, DE**

72 Inventor/es:

KRAMPOSTHUBER, GEORG

74 Agente/Representante:

ARPE FERNÁNDEZ, Manuel

ES 2 407 632 T3

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

DESCRIPCIÓN

Gestión de memoria en un soporte de datos portátil

[0001] La presente invención se refiere a un procedimiento para la gestión dinámica de una zona de memoria *heap* (*montón*) o memoria dinámica de un soporte de datos portátil, y a un soporte de datos portátil con una gestión de memoria de este tipo.

[0002] Los soportes de datos portátiles, por ejemplo tarjetas chip, tarjetas de telefonía móvil o similares disponen de pocos recursos de memoria volátil debido a su diseño, por ejemplo en una memoria de trabajo RAM volátil, a la que una gestión de memoria del soporte de datos le asigna los procesos correspondientes ejecutados de forma concurrente. La memoria libre requerida por un proceso durante el tiempo de ejecución se pone a disposición de éste en una zona de memoria *heap* lo más coherente posible de la memoria volátil, y la memoria ya no requerida por un proceso se libera para que a continuación esté disposición de otros procesos como memoria libre. Por consiguiente, los procesos concurrentes compiten por la memoria libre limitada de la zona de memoria *heap*.

[0003] La asignación y liberación continua de memoria libre por la gestión de memoria implica una fragmentación progresiva de la zona de memoria *heap*, pero ésta se puede evitar con una desfragmentación o compactación. Por otro lado es necesario encontrar y liberar en la zona de memoria *heap* aquellos objetos de memoria que ya no son requeridos por los procesos correspondientes.

[0004] Normalmente, estas dos tareas son realizadas por una depuración automática ("Garbage Collection" - recuperación de espacio inutilizado) de la zona de memoria *heap*, que es ejecutada regularmente o en caso necesario por la gestión de memoria del soporte de datos. Sin embargo, dado que el alcance de la depuración de memoria depende entre otras cosas del tamaño y el grado de fragmentación de la zona de memoria *heap*, por regla general no se puede prever el momento ni la duración de la depuración de memoria. La depuración de memoria también puede interrumpir la ejecución de un proceso, por ejemplo cuando un proceso requiere memoria libre que previamente se ha de obtener mediante una depuración de memoria. Una interrupción imprevisible de este tipo de la ejecución de un proceso no es deseable, sobre todo en caso de procesos con operaciones con tiempos de ejecución críticos, como por ejemplo en la comunicación de un soporte de datos portátil con un dispositivo de lectura o similares. Sin embargo, estas situaciones se producen a menudo en las zonas de memoria *heap* limitadas de los soportes de datos portátiles, de modo que en estos casos se requieren frecuentes depuraciones de memoria.

[0005] En este contexto, el documento US 2001/0023478 A1 da a conocer una depuración de memoria interrumpible, que puede continuar y finalizar después de la interrupción recurriendo a informaciones de estado de la depuración de memoria interrumpida. El documento WO 2005/045683 también da a conocer una depuración de memoria que puede interrumpirse, pero ésta se puede ejecutar por secciones, por ejemplo dentro del intervalo de tiempo de respuesta disponible para la respuesta a una instrucción externa. De este modo se evitan errores de comunicación debidos a demoras de respuesta. Sin embargo, esta interrupción y continuación de una depuración de memoria requiere estructuras y procesos de gestión especiales, que suponen una carga adicional para los recursos limitados de un soporte de datos portátil.

[0006] Además, el documento DE 10040974 A1 da a conocer una ampliación virtual de una zona de memoria de pila volátil de un soporte de datos portátil mediante la transferencia de objetos de la memoria de pila a una memoria no volátil. Pero esto no soluciona el problema del retardo de operaciones de proceso con tiempos de ejecución críticos por una depuración de memoria en la zona de memoria *heap*.

[0007] El documento WO 99/64955 A1 da a conocer un procedimiento de *garbage collection* que se desarrolla en una memoria *heap*, en el que la memoria *heap* se expande y contrae dinámicamente, es decir aumenta y disminuye de tamaño, correspondientemente al espacio de memoria necesario.

[0008] El documento US 2007/0192388 A1 da a conocer un procedimiento para la depuración de memoria en el que solo se realiza una *garbage collection* de una memoria *heap* una vez ejecutados los hilos (*threads* [operaciones de proceso]) en tiempo real, correspondientemente al preámbulo de la reivindicación 1.

[0009] Por consiguiente, el objetivo de la presente invención consiste en proponer un procedimiento para la gestión dinámica de una zona de memoria *heap*, que evite las desventajas de la depuración automática de memoria arriba mencionadas.

[0010] Este objetivo se resuelve según la invención mediante un procedimiento y un soporte de datos portátil con las características indicadas en las reivindicaciones independientes. Las reivindicaciones subordinadas a éstas describen configuraciones ventajosas y perfeccionamientos de la invención.

[0011] La presente invención se basa en la idea fundamental de suspender o desactivar la depuración automática de memoria en el soporte de datos durante el tiempo de ejecución de una operación de proceso con tiempo de ejecución crítico, de tal modo que la depuración de memoria no se inicie durante el tiempo de ejecución de la operación de proceso correspondiente. Durante la desactivación de la depuración de memoria, si la operación de proceso correspondiente requiere memoria libre con un tamaño que no puede ser proporcionado de momento por la

zona de memoria *heap*, la memoria libre requerida se proporciona de otro modo (es decir, no mediante una depuración de memoria) en un intervalo de tiempo tan corto que no influye negativamente en el comportamiento de tiempo de ejecución de la operación de proceso con tiempo de ejecución crítico. En particular, la memoria libre es asignada a la operación de proceso con tiempo de ejecución crítico en un intervalo de tiempo más corto que el que sería posible mediante la depuración de memoria.

[0012] Esta puesta a disposición de memoria libre en un plazo breve para operaciones de proceso con tiempos de ejecución críticos, en un modo de tiempo de ejecución optimizado alternativo a la depuración automática de memoria usual, está configurada en particular de tal modo que la memoria libre requerida puede ser ponerse a disposición en un intervalo de tiempo previsible con suficiente exactitud, de modo que, en cuanto la operación de proceso con tiempo de ejecución crítico transmite una petición de memoria libre, ya existe la seguridad de que la asignación puede tener lugar a tiempo de modo que no resulte perjudicado el comportamiento de tiempo de ejecución de la operación de proceso.

[0013] A diferencia de la depuración automática de memoria, que puede conducir a interrupciones (en gran medida imprevisibles) de una operación de proceso, ya que la duración de la interrupción depende de parámetros en los que no se puede influir sin más, como por ejemplo el tamaño de la zona de memoria *heap* a depurar, el grado de fragmentación o los objetos de memoria que ya no son necesarios, el intervalo de tiempo para la proporcionar memoria libre en el marco del modo de tiempo de ejecución optimizado está por debajo de un umbral crítico a partir del cual es de esperar un perjuicio del comportamiento de tiempo de ejecución de una operación de proceso con tiempo de ejecución crítico. Esto no implica ningún gasto de gestión especial para la gestión de memoria o para un sistema operativo del soporte de datos.

[0014] Un procedimiento según la invención en un soporte de datos portátil con una zona de memoria *heap* gestionada dinámicamente en una memoria volátil, en el que una gestión de memoria para proporcionar memoria libre puede llevar a cabo una depuración automática de la zona de memoria *heap*, consiste en las siguientes etapas:

- desactivación de la depuración automática de la zona de memoria *heap* mediante la gestión de memoria durante la realización de una operación de proceso con tiempo de ejecución crítico por un procesador del soporte de datos, y activación de un modo de tiempo de ejecución optimizado;

- puesta a disposición de memoria libre para la operación de proceso con tiempo de ejecución crítico sin influir negativamente en la ejecución de la operación de proceso con tiempo de ejecución crítico, si la operación de proceso emite una petición de memoria libre en el marco del modo de tiempo de ejecución optimizado;

- desactivación del modo de tiempo de ejecución optimizado y activación de la depuración automática de memoria una vez finalizada la operación de proceso con tiempo de ejecución crítico.

[0015] Correspondientemente, un soporte de datos según la invención incluye un procesador, una memoria volátil con una zona de memoria *heap* dispuesta dentro de ésta, un sistema operativo con una gestión de memoria, y al menos una aplicación ejecutable como proceso en el procesador, pudiendo ejecutar la gestión de memoria una depuración automática de la zona de memoria *heap* regularmente o en caso necesario con el fin de lograr suficiente memoria libre para asignarla a operaciones de proceso. En relación con la ejecución de operaciones de proceso con tiempo de ejecución crítico, la gestión de memoria está preparada para desactivar la depuración automática de memoria durante la ejecución de la operación de proceso con tiempo de ejecución crítico y conmutar a un modo de tiempo de ejecución optimizado, y, al recibir una petición de memoria libre en el modo de tiempo de ejecución optimizado, proporciona la memoria libre solicitada en un intervalo de tiempo que no influye negativamente en la ejecución de la operación de proceso con tiempo de ejecución crítico.

[0016] La memoria libre solicitada en el modo de tiempo de ejecución optimizado se proporciona incluyendo una zona de memoria fuera de la zona de memoria *heap*, por ejemplo incluyendo un área de la memoria volátil diferente de la zona de memoria *heap* o un área de una memoria del soporte de datos diferente de la memoria volátil. La facilitación de memoria libre en el modo de tiempo de ejecución optimizado tiene lugar mediante operaciones simples de escritura/lectura en esta área de memoria diferente de la zona de memoria *heap*. Dichas operaciones de escritura/lectura son en gran medida independientes del tamaño de la zona de memoria correspondiente (diferente de la zona de memoria *heap*) o de su grado de fragmentación, y pueden ser ejecutadas en un intervalo de tiempo prácticamente constante y previsible, que depende esencialmente de la velocidad de escritura/lectura de la zona de memoria correspondiente.

[0017] De acuerdo con la invención, la gestión de memoria puede proporcionar la memoria libre solicitada en el modo de tiempo de ejecución optimizado por un lado en un área de la memoria volátil diferente de la zona de memoria *heap* y, por otro lado, en un área de una memoria del soporte de datos diferente de la memoria volátil. Por consiguiente, la memoria libre solicitada puede ser proporcionada a la memoria permanente del soporte de datos por ejemplo en una zona de memoria de pila de la memoria volátil o en una zona de memoria *heap* permanente. De este modo se establece una zona de memoria *heap* virtual que incluye la zona de memoria *heap* regular (volátil) y la zona de memoria diferente de la zona de memoria *heap* en la memoria volátil o en la memoria del soporte de datos diferente de la memoria volátil.

[0018] Además, la memoria libre solicitada en el modo de tiempo de ejecución optimizado se pone a disposición en la zona de memoria *heap* transfiriendo uno o más objetos de memoria almacenados en la zona de memoria *heap* a una zona de memoria (diferente de la zona de memoria *heap*) de la memoria volátil o de una memoria del soporte de datos diferente de la memoria volátil, y asignando la memoria libre obtenida de este modo en la zona de memoria *heap*. Preferentemente, la memoria libre solicitada se pone a disposición en la zona de memoria *heap* mediante transferencia de un objeto de memoria a la zona de memoria de pila de la memoria volátil, o mediante transferencia a una zona de memoria *heap* permanente de la memoria permanente del soporte de datos. De hecho, en esta forma de realización también se establece una zona de memoria *heap* virtual que consiste en la zona de memoria *heap* existente y la zona de memoria (diferente de la zona de memoria *heap*) a la que se han transferido el objeto o los objetos de memoria.

[0019] De acuerdo con una forma de realización especialmente preferente, la gestión de memoria proporciona la memoria libre de tal modo que un espacio de memoria libre solicitado en el modo de tiempo de ejecución optimizado solo se proporciona mediante transferencia de uno o más objetos de memoria de la zona de memoria *heap* (de acuerdo con la segunda forma de realización) cuando la memoria libre solicitada no puede ser asignada de forma coherente en el área de la memoria volátil diferente de la zona de memoria *heap* o en la memoria del soporte de datos diferente de la memoria volátil (de acuerdo con la primera forma de realización). Por consiguiente, en caso de una petición de memoria libre de una operación de proceso con tiempo de ejecución crítico, la gestión de memoria comprueba en primer lugar si la zona de memoria de pila de la memoria volátil o una zona de memoria *heap* permanente de la memoria permanente del soporte de datos presentan un espacio de memoria libre coherente suficientemente grande. Si es así, la memoria libre es asignada a la operación de proceso que la solicita. De lo contrario, uno o más de los objetos de memoria presentes en la zona de memoria *heap* se transfieren a la zona de memoria de pila o a la zona de memoria *heap* permanente, de tal modo que en la zona de memoria *heap* volátil queda un espacio de memoria libre coherente suficientemente grande que puede ser asignado a la operación de proceso que lo solicita.

[0020] La selección del objeto de memoria a transferir por la gestión de memoria se puede basar en heurísticas predeterminadas. Preferentemente, la gestión de memoria selecciona un único objeto de memoria cuyo tamaño (en caso dado junto con eventuales zonas de memoria libre contiguas) corresponde al menos al tamaño de la memoria libre solicitada. Preferiblemente se selecciona un objeto de memoria cuyo tamaño corresponde a un múltiplo de una página de memoria de la zona de memoria diferente de la zona de memoria *heap*, ya que de este modo se asegura la transferencia más eficiente y rápida posible.

[0021] La gestión de memoria marca adecuadamente o registra de otro modo la memoria libre que queda en la zona de memoria *heap* después de una asignación en la misma de la memoria libre solicitada, para poder proporcionar posteriormente memoria libre en el modo de tiempo de ejecución optimizado en caso de otra petición de memoria libre por una operación de proceso con tiempo de ejecución crítico.

[0022] Si la gestión de memoria, para proporcionar la memoria libre solicitada en el modo de tiempo de ejecución optimizado, establece una memoria *heap* virtual que amplía la zona de memoria *heap* existente en la memoria volátil con partes de otra zona de la memoria volátil, por ejemplo de la zona de memoria de pila, o con una zona de memoria *heap* permanente, en dicha memoria *heap* ampliada se establece preferentemente una estructura de memoria *heap* de tal modo que un objeto de memoria transferido perteneciente a un proceso está disponible para la ejecución del proceso por el procesador del mismo modo que un objeto de memoria correspondiente presente en la zona de memoria *heap* volátil. Por consiguiente, en la memoria *heap* virtual total se establece una estructura de memoria *heap* lógica transparente.

[0023] Preferentemente, la gestión de memoria, o un dispositivo de control del soporte de datos en conexión con la gestión de memoria, detecta cuándo se inicia una operación de proceso con tiempo de ejecución crítico para desactivar inmediatamente la depuración automática de memoria y conmutar al modo de tiempo de ejecución optimizado. Del mismo modo, preferentemente también se detecta de forma automática el momento en el que finaliza la operación de proceso con tiempo de ejecución crítico correspondiente. La gestión de memoria o el dispositivo de control correspondiente detecta automáticamente la presencia de condiciones de tiempo de ejecución predeterminadas de una operación de proceso con tiempo de ejecución crítico, por ejemplo la comunicación a través de una interfaz sin contacto, la ejecución de instrucciones con tiempo de ejecución crítico, o similares. Igualmente es posible que la gestión de memoria no reconozca de forma autónoma una operación de proceso con tiempo de ejecución crítico, sino que la desactivación de la depuración automática de memoria y la conmutación al modo de tiempo de ejecución optimizado sean ordenadas externamente. Por ejemplo, pueden ser ordenadas por otros procesos de aplicación o rutinas del sistema operativo, ejecutados de forma concurrente a través de una interfaz de programación adecuada de la gestión de memoria.

[0024] Otras características y ventajas de la invención se desprenden de la siguiente descripción de ejemplos de realización según la invención y otras alternativas de realización con referencia a los dibujos adjuntos, en los que:

- la figura 1 muestra una forma de realización preferente de un soporte de datos según la invención;
- la figura 2 muestra una disposición de memoria según la invención en una zona de memoria *heap* virtual; y

- la figura 3 muestra un diagrama de operaciones esquemático de un procedimiento preferente según la invención en un soporte de datos conforme a la figura 1.

[0025] Con referencia a la figura 1, un soporte de datos portátil 10, que en este caso está configurado como una tarjeta chip, incluye una interfaz de comunicación de datos 20, un procesador (CPU) 30 y diferentes memorias 40, 50 y 60. El soporte de datos 10 también puede estar configurado de otro modo, por ejemplo tal como un testigo USB, una tarjeta multimedia segura, una tarjeta de telefonía móvil o similar.

[0026] La interfaz de comunicación de datos 20 está configurada como un campo de contactos según ISO 7816 y sirve para la comunicación con un dispositivo de procesamiento de datos externo (no representado), por ejemplo un terminal de tarjetas, y como fuente de alimentación para el soporte de datos 10. Alternativamente, el soporte de datos 10 también puede disponer de una fuente de alimentación propia (no representada), por ejemplo a través de una batería. También pueden estar previstas otras interfaces por contacto. Alternativa o adicionalmente, también puede estar prevista una interfaz de comunicación de datos sin contacto (no representada) para una comunicación de datos sin contacto, por ejemplo una antena.

[0027] En la memoria ROM permanente no regrabable 40 está grabado un sistema operativo multitarea o *multithread* 42 que controla el soporte de datos 10, preferentemente basado en Java o tarjeta Java o representado o complementado mediante una máquina virtual. El sistema operativo 42 incluye una gestión de memoria 44 para gestionar los recursos de memoria del soporte de datos 10, en particular para gestionar objetos de memoria almacenados en la memoria RAM volátil 50 para el tiempo de ejecución de los procesos ejecutados de forma concurrente en el procesador 30. La memoria RAM 50 sirve como memoria volátil principal para el soporte de datos 10.

[0028] Alternativamente, el sistema operativo 42 o partes del mismo, como por ejemplo la gestión de memoria 44, también pueden estar almacenados en la memoria *flash* regrabable 60, que alternativamente también puede estar configurada como memoria EEPROM o similar. En la memoria *flash* 60 está almacenada una aplicación 64 que puede ser ejecutada por el procesador 30 como un proceso de aplicación que puede abarcar varios *threads* (denominados "procesos de poco peso") ejecutados de forma concurrente, que procesan a su vez determinadas operaciones de proceso. Por regla general, la memoria *flash* 60 sirve para el almacenamiento permanente de datos en el soporte de datos 10, tales como los que se producen por ejemplo durante la ejecución de procesos de aplicación.

[0029] Durante la generación y ejecución de procesos de aplicación mediante el procesador 30, la gestión de memoria 44 asigna recursos de memoria de la memoria RAM volátil 50 a los procesos/*threads* individuales. Para los objetos de memoria generados dinámicamente por un proceso/*thread* durante su tiempo de ejecución, la gestión de memoria 44 gestiona una zona de memoria *heap* 51 (V-heap) en la memoria volátil 50. Esta área de memoria *heap* volátil 51 también está indicada en la figura 2. Las áreas sombreadas mostradas en la figura 2 representan las secciones de memoria de la zona de memoria *heap* volátil 51 ocupadas por objetos de memoria de procesos ejecutados.

[0030] En la zona de memoria de pila 52 (*stack*), la gestión de memoria 44 gestiona memorias de pila (*stack*) para los procesos ejecutados de forma concurrente. Los tamaños de las zonas de memoria *heap* y de pila 51, 52 no están limitados por especificaciones estáticas, es decir, la gestión de memoria 44 gestiona dinámicamente las dos zonas de memoria en respuesta a peticiones de memoria libre de los procesos ejecutados. Como muestra la figura 2, en este contexto puede ocurrir que un proceso solicite un espacio de memoria libre A con un tamaño que, si bien no está disponible de forma coherente en la zona de memoria *heap* volátil 51, sí está presente como la suma de varias zonas de memoria libre más pequeña. Si el proceso o la operación de proceso que solicita memoria libre no presenta un tiempo de ejecución crítico, el espacio de memoria libre A solicitado puede ser proporcionado mediante una depuración de memoria convencional de la zona de memoria *heap* volátil 51, en la que una compactación por "desplazamiento de aproximación" crea un espacio de memoria libre coherente suficientemente grande. Dado que en este proceso se depura o compacta toda la zona de memoria *heap* volátil 51, la depuración de memoria es un procedimiento que requiere un tiempo de procesamiento relativamente largo, que puede llegar a ser de algunos cientos de ms.

[0031] Sin embargo puede ocurrir que, durante la ejecución de la aplicación 64 como proceso de aplicación, determinadas operaciones de este proceso o sus *threads* presenten tiempos de ejecución especialmente críticos y que, para lograr un procesamiento con éxito, dichas operaciones no puedan ser interrumpidas por la depuración automática de memoria. Estas operaciones de proceso con tiempos de ejecución críticos están relacionadas principalmente con procesos de comunicación con aparatos externos, por ejemplo la comunicación de datos sin contacto o por contacto a través de protocolos de comunicación de datos adecuados, o también accesos a datos de *web* estáticos.

[0032] Por ello, en caso de una petición de memoria libre de una operación de proceso con tiempo de ejecución crítico, la gestión de memoria 44 precisamente no lleva a cabo ninguna depuración automática de memoria, sino que la desactiva y efectúa una conmutación a un modo de tiempo de ejecución optimizado alternativo, en el que la petición de memoria libre de la operación de proceso con tiempo de ejecución crítico se puede satisfacer más

rápidamente que mediante la depuración de la zona de memoria *heap* volátil 51 y sin influir negativamente en el comportamiento de tiempo de ejecución del proceso con tiempo de ejecución crítico.

[0033] En el modo de tiempo de ejecución optimizado, la gestión de memoria dispone de, al menos, dos procedimientos de puesta a disposición de memoria alternativos, con los que la memoria libre deseada se puede poner rápidamente a disposición de la operación de proceso que la solicita. En un primer procedimiento de facilitación de memoria, un espacio de memoria libre coherente solicitado y no disponible en la zona de memoria *heap* 51 se establece y proporciona en la zona de memoria de pila volátil 52 o en una zona de memoria *heap* permanente 61 (P-*heap*) de la memoria *flash* 60. Por ejemplo, en la zona de memoria de pila 52 entre dos marcos de memoria de pila se puede proporcionar un espacio de memoria libre.

[0034] De acuerdo con el segundo procedimiento de puesta a disposición de memoria, la memoria libre solicitada se crea en la zona de memoria *heap* 51 transfiriendo un objeto de memoria C que se encuentra dentro de ésta a la zona de memoria de pila volátil 52 o a una zona de memoria *heap* permanente 61. Este proceso está ilustrado en la figura 2. La memoria libre solicitada A no está disponible de forma coherente en la zona de memoria *heap* 51. Por ello, un objeto de memoria C de la zona de memoria *heap* volátil 51 se transfiere a una zona de memoria libre B suficientemente grande de la zona de memoria de pila volátil 52 o de la zona de memoria *heap* permanente 61, de modo que la memoria libre D resultante en la zona de memoria *heap* 51 es suficientemente grande para satisfacer la petición de memoria libre A del proceso con tiempo de ejecución crítico.

[0035] En la transferencia de un objeto de memoria C a un espacio de memoria libre B suficientemente grande de la zona de memoria de pila 52 o de la zona de memoria *heap* permanente 61 se selecciona un objeto de memoria C de tamaño adecuado para poder satisfacer al menos la petición de memoria A de la operación de proceso con tiempo de ejecución crítico. En particular se seleccionan y transfieren objetos de memoria C que son esencialmente más grandes que la memoria libre solicitada A y que a ser posible correspondan a un múltiplo del tamaño de una página de memoria de la zona de memoria de transferencia en la memoria de pila 52 o en la memoria permanente 60, por ejemplo 128 octetos en caso de una memoria EEPROM permanente 60. De este modo, en la zona de memoria *heap* 51 se crea un espacio de memoria libre D lo más grande posible manteniendo inalterado el tiempo de programación de la memoria permanente 60. La gestión de memoria 44 marca o registra la memoria libre D - A que queda eventualmente después de la asignación de la memoria libre A en la zona de memoria libre D, y la asigna en caso de otra petición de memoria libre de una operación de proceso con tiempo de ejecución crítico.

[0036] En cada uno de los dos procedimientos de facilitación de memoria se forma una memoria *heap* virtual 53 que consiste en la zona de memoria *heap* volátil 51 más una parte en la zona de memoria de pila 52 o de la zona de memoria *heap* permanente 61, que se utiliza bien para la asignación directa a una operación de memoria con tiempo de ejecución crítico, bien para la transferencia de un objeto de memoria C desde la zona de memoria *heap* 51. En la memoria *heap* virtual total 53 se establece uniformemente una estructura de memoria *heap* lógica transparente, de modo que los objetos de memoria transferidos a la zona de memoria de pila 52 o a la zona de memoria *heap* permanente 61 están accesibles, sin más, durante la ejecución del proceso correspondiente, con lo que los objetos de memoria transferidos siguen estando disponibles en todo momento para el procesador 30, aunque con un mayor tiempo de escritura/lectura.

[0037] Mientras que una depuración automática de toda la zona de memoria *heap* 51 puede requerir unos cientos de milisegundos, el tiempo necesario para la transferencia de un objeto de memoria a la zona de memoria *heap* permanente 61 es considerablemente menor. Los tiempos de programación de una memoria EEPROM permanente oscilan entre aproximadamente 2 y 4 milisegundos para la escritura de hasta 128 octetos, o en caso de una memoria *flash* aproximadamente 8 milisegundos para la escritura de hasta 256 octetos. Sin embargo, por regla general estos tiempos de retardo no influyen negativamente en la ejecución de una operación de proceso con tiempos de ejecución críticos, de modo que ésta puede mantener los tiempos de ejecución necesarios a pesar de la asignación de nueva memoria libre.

[0038] Los dos procedimientos de facilitación de memoria descritos se basan en la obtención de nueva memoria libre mediante la inclusión de una zona de memoria fuera de la zona de memoria *heap* en un modo de tiempo de ejecución optimizado, evitando la usual depuración automática de memoria. Complementariamente, en caso necesario la gestión de memoria también puede disponer de un tercer procedimiento de facilitación de memoria, en el que se lleva a cabo una depuración de memoria parcial simplificada en lugar de la usual depuración de memoria completa. En este caso se detectan los objetos de memoria de la zona de memoria *heap* volátil 51 que ya no son requeridos por los procesos correspondientes, por ejemplo mediante el procedimiento conocido *mark-and-sweep* (marcar y barrer). La memoria ocupada por estos objetos de memoria se puede liberar entonces como memoria libre y mediante una compactación (parcial) adicional se puede crear más memoria libre coherente. Esta depuración de memoria parcial simplificada se lleva a cabo hasta que queda disponible la memoria libre solicitada, y entonces se interrumpe.

[0039] La gestión de memoria 44 lleva a cabo principalmente una combinación del primer y el segundo procedimiento de facilitación de memoria intentando en primer lugar proporcionar un espacio de memoria libre suficientemente grande en la zona de memoria de pila 52 o en la zona de memoria *heap* permanente 61 y, si esto no es posible, transfiriendo un objeto de memoria C a un espacio de memoria libre B apropiado de la zona de memoria

de pila 52 o de la zona de memoria *heap* permanente 61. La figura 3 muestra un bosquejo de un procedimiento de este tipo.

[0040] La gestión de memoria 44 bosquejada en la figura 3 abarca varios módulos de gestión: una gestión de objetos 44a, una gestión *V-heap* 44b, una gestión *stack* 44c y una gestión *P-heap* 44d.

5 **[0041]** En la etapa S1, la gestión de memoria 44 reconoce el inicio de una operación de proceso (procop) con tiempo de ejecución crítico, desactiva la depuración automática de memoria y activa el modo de tiempo de ejecución optimizado. La gestión de memoria 44 detecta requisitos de tiempo real determinados automáticamente o características de un procesamiento en tiempo real mediante una operación de proceso, por ejemplo secuencias de comunicación a través de una interfaz sin contacto o similares. Alternativamente, en la etapa S1, una rutina del sistema operativo u otro proceso de aplicación puede ordenar a la gestión de memoria 44, a través de una interfaz programable, la desactivación de la depuración usual de memoria y active el modo de tiempo de ejecución optimizado.

15 **[0042]** En la etapa S2, la operación de proceso con tiempo de ejecución crítico solicita a la gestión de objetos 44a memoria libre para un nuevo objeto de memoria (new object x byte [objeto nuevo x octetos]). En el paso S3, la gestión de objetos 44a solicita la memoria libre a la gestión *V-heap* 44b (allocate x byte [asignar x octetos]), que devuelve un mensaje de error (error) porque no hay presente suficiente memoria libre en la zona de memoria *heap* 51. En el paso S4, la gestión de objetos 44a solicita la memoria libre a la gestión *stack* 44c (allocate x byte), que también devuelve un mensaje de error (error), ya que en la zona de memoria de pila 52 tampoco hay disponible ningún espacio de memoria libre suficientemente grande. Por consiguiente, no es posible una asignación de un espacio de memoria libre de la zona de memoria de pila 52 de acuerdo con el primer procedimiento de facilitación de memoria anteriormente descrito. En la etapa S5, la gestión de objetos 44a busca en la zona de memoria *heap* volátil 51 un objeto de memoria adecuado C de tamaño suficiente (search object size [buscar tamaño objeto] y > x) y solicita en la etapa S6 un espacio de memoria libre en la zona de memoria *heap* permanente 61 con el tamaño y del objeto de memoria encontrado (allocate y byte asignar y octetos). La gestión *P-heap* 44d detecta un espacio de memoria suficientemente grande en la zona de memoria *heap* permanente 61, tras lo cual, en la etapa S7, el objeto de memoria a mover es transferido de la gestión de objetos 44a a la gestión *V-heap* 44b (transfer object [transferir objeto]), que copia el objeto de memoria en el espacio de memoria libre encontrado en la zona de memoria *heap* permanente 61 (copy object [copia objeto]). En la etapa S9, la gestión *heap* 44b marca (mark) como memoria libre el espacio de memoria libre, creado de este modo en la zona de memoria *heap* volátil 51, y en la etapa S10 lo notifica a la gestión de objetos 44a. En la etapa S11, la gestión de objetos 44a solicita memoria libre A de nuevo a la gestión *V-heap* 44b (allocate x byte), que ahora puede asignar la memoria libre de modo que la operación de proceso con tiempo de ejecución crítico puede continuar en la etapa S12.

35 **[0043]** El tratamiento según la invención de peticiones de memoria libre de operaciones de proceso con tiempos de ejecución críticos mediante la gestión de memoria 44 en el modo de tiempo de ejecución optimizado se puede utilizar en particular en la aceleración de determinadas operaciones de proceso en la implementación API, y también en caso de servicios de sistema ejecutados en código de programa nativo en la mayoría de los casos. No obstante, en principio los procedimientos descritos son adecuados para acelerar cualquier aplicación o sus procesos de aplicación, siendo también posible conectar y desconectar selectivamente el modo de tiempo de ejecución optimizado, por ejemplo a través de una interfaz de programación de acceso público de la gestión de memoria 44 que puede ser utilizada por cualquier aplicación.

REIVINDICACIONES

1. Procedimiento en un soporte de datos portátil (10) con una memoria volátil (50) y una gestión de memoria dinámica (44) de una zona de memoria *heap* (51) de la memoria volátil (50), en el que la gestión de memoria (44) puede llevar a cabo una depuración automática de la zona de memoria *heap* (51) para proporcionar memoria libre, y en el que la gestión de memoria (44), durante una ejecución de una operación de proceso con tiempo de ejecución crítico en el soporte de datos (10), desactiva la depuración automática de memoria (S1) y, a petición de la operación de proceso (S2), proporciona memoria libre dentro de un intervalo de tiempo que no influye negativamente en la ejecución de la operación de proceso con tiempo de ejecución crítico (S3-S12), **caracterizado porque** la memoria libre solicitada se proporciona incluyendo una zona de memoria (52, 61) fuera de la zona de memoria *heap* (51) (S2-S12), y porque la memoria libre solicitada se pone a disposición en la zona de memoria *heap* (51) mediante transferencia de un objeto de memoria (S7-S8) a una zona de memoria (52) de la memoria volátil (50) diferente de la zona de memoria *heap* (51), o a una zona de memoria (61) de una memoria (60) del soporte de datos (10) diferente de la memoria volátil (50) (S3-S12).
2. Procedimiento según la reivindicación 1, **caracterizado porque** la memoria libre solicitada se proporciona en una zona de memoria de pila (52) de la memoria volátil (50).
3. Procedimiento según la reivindicación 2, **caracterizado porque** la memoria libre solicitada se proporciona mediante la transferencia de un objeto de memoria a una zona de memoria (61) de la memoria permanente (60) del soporte de datos (10) (S7-S8).
4. Procedimiento según una de las reivindicaciones 1 a 3, **caracterizado porque** la memoria libre solicitada solo se proporciona mediante la transferencia de un objeto de memoria (S7-S8) cuando la memoria libre coherente presente en un espacio de memoria (52) de la memoria volátil (50) diferente de la zona de memoria *heap* (51), o en una zona de memoria (61) de una memoria (60) del soporte de datos (10) diferente de la memoria volátil (50), es menor que la memoria libre solicitada (S4).
5. Procedimiento según una de las reivindicaciones 1 a 4, **caracterizado porque** se selecciona un objeto de memoria a transferir tan grande como sea posible (S5), preferentemente un objeto de memoria cuyo tamaño corresponde a un múltiplo del tamaño de una página de memoria de la zona de memoria (52) de la memoria volátil (50) diferente de la zona de memoria *heap* (51), o de la zona de memoria (61) de una memoria (60) del soporte de datos (10) diferente de la memoria volátil (50).
6. Procedimiento según una de las reivindicaciones 1 a 5, **caracterizado porque** la memoria libre que queda en la zona de memoria *heap* (51) después de proporcionar la memoria libre solicitada se marca para poder proporcionar memoria libre en caso de otra petición de memoria libre por una operación de proceso con tiempo de ejecución crítico.
7. Procedimiento según una de las reivindicaciones 1 a 6, **caracterizado porque** en la zona de memoria (52) de la memoria volátil (50) diferente de la zona de memoria *heap* (51), o en una zona de memoria (61) de una memoria (60) del soporte de datos (10) diferente de la memoria volátil (50), se establece una estructura de memoria *heap* de tal modo que un objeto de memoria transferido perteneciente a un proceso está disponible para la posterior ejecución del proceso.
8. Procedimiento según una de las reivindicaciones 1 a 7, **caracterizado porque** la gestión de memoria (44) reconoce (S1) una operación de proceso con tiempo de ejecución crítico a ejecutar y a continuación desactiva la depuración automática de memoria hasta haberse ejecutado la operación de proceso con tiempo de ejecución crítico.
9. Procedimiento según una de las reivindicaciones 1 a 8, **caracterizado porque** un proceso de sistema o aplicación ordena a la gestión de memoria (44) la desactivación de la depuración automática de memoria.
10. Soporte de datos portátil (10) que incluye un procesador (30), una memoria volátil (50), un sistema operativo (42) con una gestión de memoria (44), y al menos una aplicación (64) ejecutable como proceso en el procesador (30), estando configurada la gestión de memoria (44) para llevar a cabo una depuración automática de una zona de memoria *heap* (51) de la memoria volátil (50) con el fin de proporcionar memoria libre, y estando configurada también la gestión de memoria para desactivar la depuración automática de memoria durante una ejecución de una operación de proceso con tiempo de ejecución crítico por el procesador (30) y para, al recibir una petición de memoria libre de la operación de proceso, proporcionar memoria libre dentro de un intervalo de tiempo que no influye negativamente en la ejecución de la operación de proceso con tiempo de ejecución crítico, **caracterizado porque** la gestión de memoria (44) está configurada además de tal modo que la memoria libre solicitada se proporciona incluyendo una zona de memoria (52, 61) fuera de la zona de memoria *heap* (51) (S2-S12), y porque la memoria libre solicitada se proporciona en la zona de memoria *heap* (51) mediante transferencia de un objeto de memoria (S7-S8) a una zona de memoria (52) de la memoria volátil (50) diferente de la zona de memoria *heap* (51), o a una zona de memoria (61) de una memoria (60) del soporte de datos (10) diferente de la memoria volátil (50) (S3-S12).

11. Soporte de datos (10) según la reivindicación 10, **caracterizado porque** la gestión de memoria (44) está preparada para ejecutar un procedimiento según una de las reivindicaciones 1 a 9.

12. Soporte de datos (10) según la reivindicación 10 u 11, **caracterizado porque** está configurado como tarjeta chip, tarjeta Java, tarjeta multimedia segura, tarjeta de telefonía móvil o medio de almacenamiento USB.

FIG 1

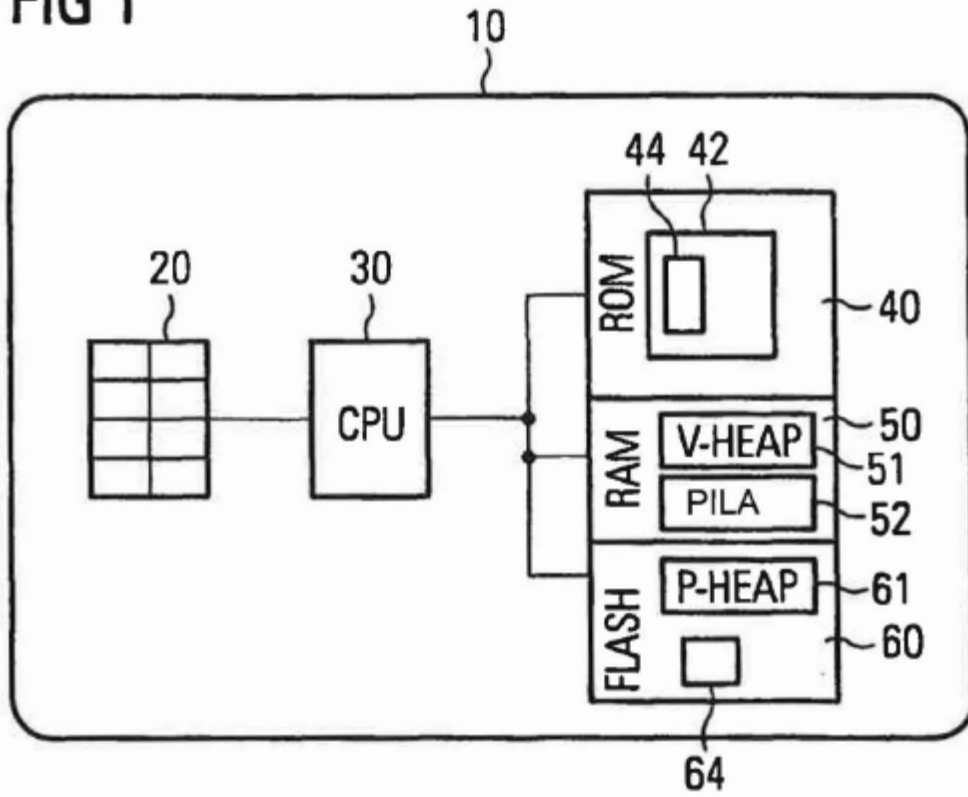


FIG 2

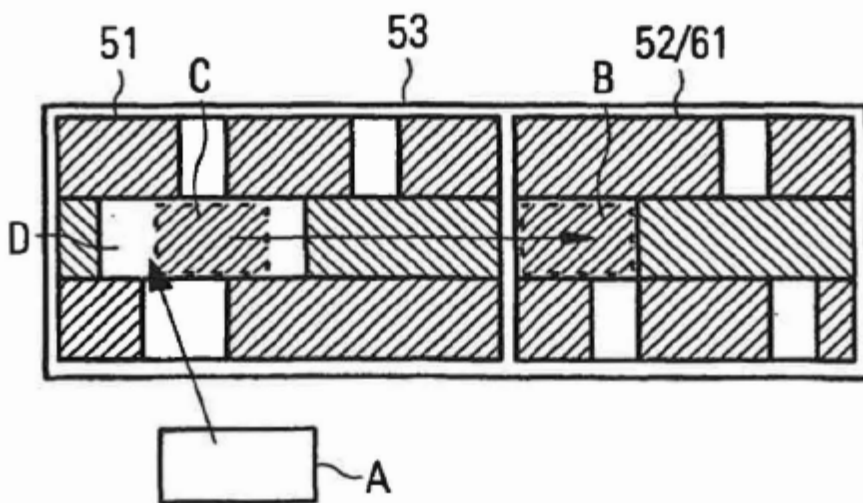
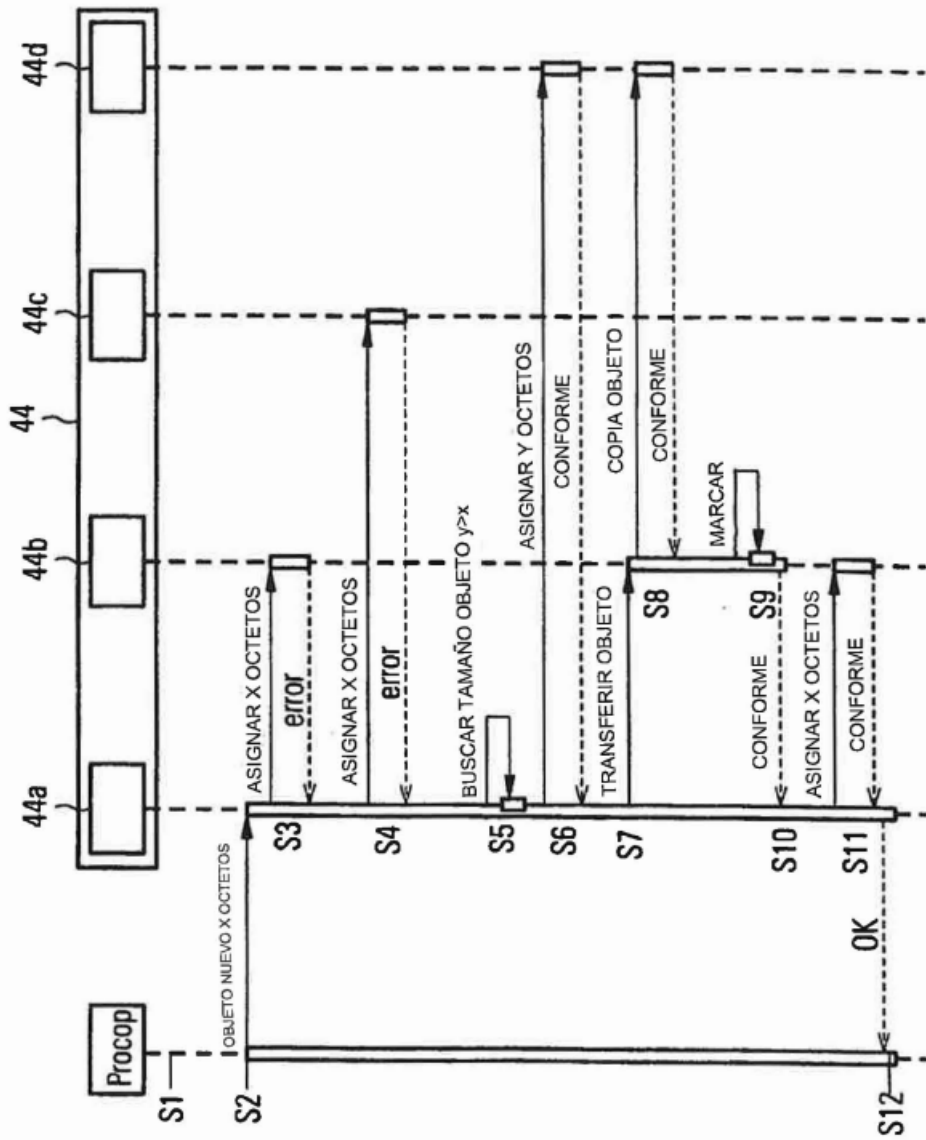


FIG 3



REFERENCIAS CITADAS EN LA DESCRIPCIÓN

La lista de referencias citada por el solicitante lo es solamente para utilidad del lector, no formando parte de los documentos de patente europeos. Aún cuando las referencias han sido cuidadosamente recopiladas, no pueden excluirse errores u omisiones y la OEP rechaza toda responsabilidad a este respecto.

5

Documentos de patente citados en la descripción

- US 20010023478 A1 [0005]
- WO 2005045683 A [0005]
- DE 10040974 A1 [0006]
- WO 9964955 A1 [0007]
- US 20070192388 A1 [0008]