(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 701 702**

(51) Int. CI.:

***G06F 21/52*** (2013.01)
***G06F 8/52*** (2008.01)
***G06F 21/51*** (2013.01)
***G06F 21/57*** (2013.01)
***G06F 9/445*** (2008.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **25.01.2017 PCT/EP2017/051476**

(87) Fecha y número de publicación internacional: **17.08.2017 WO17137256**

(96) Fecha de presentación y número de la solicitud europea: **25.01.2017 E 17702800 (8)**

(97) Fecha y número de publicación de la concesión europea: **07.11.2018 EP 3274825**

(54) Título: **Procedimiento y entorno de ejecución para la ejecución asegurada de instrucciones de programa**

(30) Prioridad:

**09.02.2016 DE 102016201898**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**25.02.2019**

(73) Titular/es:

**SIEMENS AKTIENGESELLSCHAFT (100.0%)**
**Werner-von-Siemens-Straße 1**
**80333 München, DE**

(72) Inventor/es:

**FALK, RAINER;**
**FISCHER, KAI;**
**HEINTEL, MARKUS;**
**MERLI, DOMINIK;**
**ASCHAUER, HANS;**
**KLASEN, WOLFGANG;**
**PFAU, AXEL;**
**PYKA, STEFAN y**
**SCHNEIDER, DANIEL**

(74) Agente/Representante:

**LOZANO GANDIA, José**

## DESCRIPCIÓN

### PROCEDIMIENTO Y ENTORNO DE EJECUCIÓN PARA LA EJECUCIÓN ASEGURADA DE INSTRUCCIONES DE PROGRAMA

La invención se refiere a un procedimiento y un entorno de ejecución para la ejecución asegurada de instrucciones de programa de una aplicación.

En los modernos sistemas de automatización para el control se usan sistema TI y aplicaciones. En el curso de la digitalización se desprenden soluciones individuales muy especializadas mediante sistemas multiobjetivo genéricos. Estos sistemas multiobjetivo se adaptan mediante una configuración dinámica (de la aplicación) a la finalidad de uso correspondiente. La funcionalidad no usada para la finalidad de uso especial permanece a este respecto en el sistema. En principio la funcionalidad superflua / desusada representada un posible riesgo. La funcionalidad se puede ejecutar (mediante manipulación) de forma intencionada o no intencionada y conducir a un estado del sistema indeseado. Un ejemplo es el error heartbleed con SSL/TSL que ha estado oculto mucho tiempo en una parte no necesaria típicamente de la biblioteca SSL/TSL afectada. Por tanto la función estaba a disposición en la mayoría de los sistemas y se pudo usar.

Por el estado de la técnica se conocen el documento US 8,531,247 B2, el documento US 8,892,616 B2, el documento US 8,300,811 B2, el documento US 9,147,088 B2, el documento EP 2 605 445 B1, el documento EP 2 870 565 A1, el documento EP 2 891 102 A1 y el documento US 8 843 761 B2.

Adicionalmente se conoce la solicitud de patente US 2013/305228 A1 que da a conocer un procedimiento para la selección de algoritmos y funciones de una biblioteca.

Adicionalmente se conoce la solicitud de patente US 2005/039164 A1 que da a conocer un procedimiento para la edición del código de programa.

Adicionalmente se conoce la solicitud de patente US 2011/271344 A1 que da a conocer un dispositivo de identificación de módulos dañados para la identificación y desactivación de módulos maliciosos, que trabajan en un dispositivo de procesamiento de información.

El objetivo de la presente invención es proporcionar un procedimiento y un entorno de ejecución para la ejecución asegurada de instrucciones de programa de una aplicación, que se puede realizar con un bajo coste.

El objetivo se consigue mediante las características indicadas en las reivindicaciones independientes. En las reivindicaciones dependientes están representados perfeccionamientos ventajosos de la invención.

Según un primer aspecto, la invención, para la ejecución asegurada y asistida por ordenador de instrucciones de programa de una aplicación, se refiere a un procedimiento con las etapas del procedimiento siguientes:
En una etapa del procedimiento se enciende un módulo de aprendizaje de un entorno de ejecución.

En otra etapa del procedimiento, la aplicación se ejecuta en el entorno de ejecución mientras que el modo de aprendizaje está encendido, en donde las instrucciones de programa de la aplicación se realizan para un escenario de aplicación seleccionado predeterminado y el entorno de ejecución les asocia una primera información de validez específica al escenario de la aplicación a las instrucciones de programa realizadas.

En otra etapa del procedimiento se enciende un modo de trabajo del entorno de ejecución, en donde en el modo de trabajo el entorno de ejecución verifica la primera información de validez de las instrucciones de programa, y en donde el entorno de ejecución ejecuta las instrucciones de programa en función de su información de validez.

En relación con la solicitud de patente, bajo una aplicación se puede entender un fichero ejecutable o también una biblioteca de programas.

En realización con la solicitud de patente, bajo un entorno de ejecución se puede entender una máquina virtual, por ejemplo, una máquina virtual de Java , un procesador o un entorno de sistema operativo. Un entorno de ejecución puede estar realizado en una unidad de cálculo física (procesador, microcontrolador, CPU, núcleo de CPU). A este respecto, la ejecución de la aplicación se puede realizar en un modo de aprendizaje y en un modo de ejecución en la misma unidad de cálculo física. Asimismo es posible, por ejemplo, que la ejecución de la aplicación se realice en un modo de aprendizaje en otra unidad de cálculo física. Así se puede realizar p. ej. el aprendizaje, por ejemplo, en una unidad de cálculo de aprendizaje especial. La ejecución en un modo de ejecución se realiza, por ejemplo, en una segunda unidad de cálculo, en donde a este respecto la información de validez determinada durante el aprendizaje se usa durante la ejecución en la unidad de cálculo de ejecución. La información de validez determinada, por ejemplo, mediante la unidad de cálculo de aprendizaje se proporciona preferentemente de forma protegida frente a la manipulación.

En relación con la solicitud de patente, bajo instrucciones de programa se pueden entender las instrucciones de programa que comprende una aplicación como un todo preferentemente inclusive de las bibliotecas usadas.

En relación con la solicitud de patente, bajo rutas de ejecución se pueden entender las zonas parciales de la aplicación, que comprenden varias instrucciones de programa a ejecutar directamente sucesivamente y que en particular están asociadas a una funcionalidad parcial determinada de la aplicación. Las rutas de ejecución también pueden ser zonas parciales que se ejecutan en función de una condición, por ejemplo, condición "if-else".

En relación con la solicitud de patente, bajo información/informaciones de validez se pueden entender una primera información de validez y/o una segunda información de validez y/o una tercera información de validez.

En relación con la solicitud de patente, bajo un equipo se puede entender, por ejemplo, un equipo control para un sistema de extinción de incendios, un equipo de supervisión para la supervisión de picos de tensión en alta tensión o un equipo de campo.

En relación con la solicitud de patente, bajo un escenario de aplicación o un escenario de aplicación predeterminado seleccionado se puede entender un escenario de aplicación para una aplicación que solo sirve para una finalidad determinada. Esto puede ser, por ejemplo, un equipo de control (en el que se ejecuta la aplicación), que supervisa los picos de tensión en una central eléctrica y eventualmente interviene de forma controlada. A este respecto pueden estar establecidos exactamente los componentes de hardware, con los que se comunica el equipo de control, y el entorno de uso. Esto puede significar que a la aplicación sólo se le transmiten, por ejemplo, valores de entrada con determinados rangos de valores esperados, y también sólo se usan funciones, o sus instrucciones de programa, que son necesarias por ejemplo para una supervisión de un rango de tensión determinado a alta tensión. Las funciones, que sólo se usarían para baja tensión, no se realizarían o usarían en un escenario de uso seleccionado predeterminado para alta tensión.

En relación con la solicitud de patente, bajo "sin información de validez» se puede entender que a una instrucción de programa y/o a una subrutina / función y/o a una biblioteca de programas no está asociada una información de validez. Si, por ejemplo, a cada instrucción de programa se le asocia una zona de almacenamiento determinada, en tanto que la información de validez se puede depositar, entonces para una instrucción de programa sin información de validez esta zona de memoria se puede ocupar, por ejemplo, con ceros u otro valor, que indica que para esta instrucción de programa no está disponible una información de validez. Bajo "sin información de validez" se puede entender adicionalmente que a una instrucción de programa se le asocia una información de invalidez.

Mediante el procedimiento se consigue una elevada seguridad para las aplicaciones que se ejecutan, por ejemplo, en un equipo determinado, por ejemplo un equipo de campo, dado que se suprime una ejecución de órdenes de programa sin información de validez. Esto significa, por ejemplo, que las instrucciones no realizadas resultan sin información de validez, es decir, no presentan información de validez. Antes de que se active el modo de aprendizaje, se pueden borrar opcionalmente, por ejemplo, informaciones de validez presentes, de modo que antes del encendido del modo de aprendizaje no esté asociada un información de validez a ninguna instrucción de programa.

Por ejemplo, también es posible que el encendido del modo de trabajo esté protegido por un mecanismo de seguridad, por ejemplo, una contraseña o un código de conexión o un procedimiento criptográfico, en particular en forma de un certificado digital o una estructura de ficheros de licencia.

Por ejemplo, también son concebibles distintos modos de funcionamiento. Por ejemplo, así es posible que el modo de aprendizaje y el modo de trabajo se puedan ejecutar en paralelo, para que las instrucciones de programa realizadas se detecten lo más completamente posible para el escenario de aplicación. Pero por ejemplo también es posible que mediante el encendido del modo de trabajo se desconecte automáticamente el modo de aprendizaje. Si por el contrario se enciende el modo de aprendizaje, el modo de trabajo se desconecta automáticamente. Esto se puede configurar preferentemente según el comportamiento deseado de los modos de funcionamiento del entorno de trabajo.

En una primera forma de realización del procedimiento, las instrucciones de programa realizadas se asocian a rutas de ejecución recorridas de la aplicación y una segunda información de validez específica al escenario de la aplicación se asocia respectivamente a una ruta de ejecución.

El tiempo de ejecución del procedimiento se puede mejorar dado que sólo a la ruta de ejecución, por ejemplo, en una condición de "If-Else" se le asocia una información de validez.

En otras formas de ejecución del procedimiento, la primera información de validez se asocia durante una primera fase de aprendizaje y durante una segunda fase de aprendizaje mediante el entorno de ejecución se asocia una tercera información de validez específica al escenario de la aplicación a las instrucciones de programa realizadas.

De este modo es posible mejorar la asignación de las informaciones de validez. Esto se puede realizar, por ejemplo, dado que la primera fase de aprendizaje se lleva a cabo cuando los test de funcionamiento se efectúan durante la fabricación de un equipo en el que se debe ejecutar la aplicación. La segunda fase de aprendizaje se puede ejecutar

luego en el cliente, en el que el aparato y la aplicación ejecuta el escenario de aplicación seleccionado predeterminado, por ejemplo, un control de sobrecarga en una red eléctrica de suministro. De este modo es posible reducir la duración de la detección de la información de validez en el cliente.

En otras formas de realización del procedimiento, como entorno de ejecución se usa un procesador y/o una máquina virtual y/o un núcleo de sistema operativo o un núcleo de sistema operativo usando una unidad de gestión de memoria.

Según el caso de aplicación se puede realizar de este modo una variante a implementar de forma más sencilla, en tanto que por ejemplo a las instrucciones de programa en bytecode de una aplicación de Java se les asocia una información de validez. Se consigue una seguridad muy elevada durante la ejecución de la aplicación, en tanto que en un procesador se le asocia respectivamente a una instrucción de máquina una información de validez.

Pero también es posible asociar a instrucciones de CPU de la aplicación (también se puede denominar binario) una información de validez, a fin de conseguir por ejemplo una seguridad media.

Por ejemplo, es posible en general que la información de validez se integre directamente en el entorno de ejecución, en tanto que para ello está prevista una memoria especial o zona de memoria. Pero la información de validez también puede estar depositada fuera del entorno de ejecución, por ejemplo como fichero, que está protegido preferentemente de forma criptográfica. El entorno de ejecución accede luego en el tiempo de ejecución del programa a las informaciones de validez en el fichero. Para ello puede ser necesario que el entorno de ejecución disponga eventualmente de las claves criptográficas correspondientes.

En otras formas de realización del procedimiento, mediante un disparador se borran la primera información de validez y/o la segunda información de validez y/o la tercera información de validez de las instrucciones de programa.

En situaciones, en las que se ha usado, por ejemplo, un equipo de control para la supervisión de alta tensión de un primer rango de tensión y ahora se debe usar para la supervisión de alta tensión de un segundo rango de tensión, es razonable que las funciones ya no necesarias (o sus instrucciones de programa) para la supervisión del primer rango de tensión ya no presentan una información de validez. La información de validez se puede borrar eventualmente de forma sencilla mediante el disparador, que está protegido preferentemente por un mecanismo de seguridad, por ejemplo, una contraseña o un procedimiento criptográfico.

En otras formas de realización del procedimiento, la primera información de validez y/o la segunda información de validez y/o la tercera información de validez se almacenan de forma protegida por seguridad.

Las informaciones de validez se pueden almacenar, por ejemplo, de forma protegida por seguridad, en tanto que se constituyen procedimientos criptográficos, como una codificación simétrica, codificación asimétrica u otra firma digital sobre las informaciones de validez, a fin de que se pueda verificar su integridad. También se puede conseguir un almacenamiento protegido por seguridad dado que el equipo, por ejemplo un equipo de control para la supervisión de alta tensión, en el que se ejecuta la aplicación, se puede sellar de modo que un módulo de memoria, en el que están depositadas las informaciones de validez, tampoco se puede alcanzar espacialmente (asegurado espacialmente). Esto tiene la ventaja de que de nuevo se eleva la seguridad del procedimiento .

En otras formas de realización del procedimiento, el encendido del modo de aprendizaje está protegido por un mecanismo de seguridad.

Dado que el encendido del modo de aprendizaje o también del modo de trabajo está protegido por un mecanismo de seguridad, se consigue una seguridad más elevada del procedimiento. El modo de seguridad se puede implementar, por ejemplo, mediante procedimientos criptográficos o el encendido del modo de aprendizaje sólo es posible en instantes y/o situaciones predefinidos. Esto puede ser, por ejemplo, durante la fabricación de un equipo en el que se ejecuta la aplicación. Aquí existe, por ejemplo, un acceso a una unidad de memoria sellable, que no es accesible durante un funcionamiento normal (en el modo de trabajo en particular en el lugar de uso del equipo). Un instante predefinido también puede ser, por ejemplo, una fase para test del sistema o durante la depuración (debugging) de la aplicación en el equipo.

En otras formas de realización del procedimiento, la primera información de validez y/o la segunda información de validez y/o la tercera información de validez se le proporcionan a otro equipo.

De este modo se pueden transmitir de manera lo más sencilla posible las informaciones de validez para las aplicaciones con el mismo escenario de uso. Si en una central eléctrica se usan varios equipos con la aplicación, a fin de supervisar por ejemplo de manera idéntica los picos de tensión en alta tensión en distintos puntos, se puede detectar en primer lugar la información de validez para las instrucciones de programa para un equipo y su aplicación y transmitirse la información de validez, por ejemplo, a los otros equipos. Para la transmisión pueden estar protegidas las informaciones de validez de nuevo de forma criptográfica y la transmisión se podría realizar preferentemente de forma automática.

En otras formas de realización del procedimiento, la ejecución de la aplicación en el entorno de ejecución con el modo de aprendizaje encendido se realiza en un equipo y/o en un equipo de test igual constructivamente y/o un entorno de simulación del equipo.

De este modo se pueden detectar las informaciones de validez de la manera más sencilla posible.

En otras formas de realización del procedimiento, la primera información de validez y/o la segunda información de validez y/o la tercera información de validez se asocian a las instrucciones de programa bajo la forma de instrucciones y/o de subrutinas y/o de bibliotecas.

Dado que la información de validez se puede asociar de forma flexible, la información de validez no se asocia, por ejemplo, a cada línea individual en el código de programa de la aplicación, sino que una información de validez también se le puede asociar de forma condicionada a la situación a una subrutina / función o una biblioteca de programas. Esto tiene la ventaja de que se mejora, por ejemplo, el tiempo de ejecución de la aplicación, dado que para cada línea de programa no se debe evaluar una información de validez.

En otras formas de realización del procedimiento, a las instrucciones de programa, que dependen de las instrucciones de programa con la primera información de validez y/o la segunda información de validez y/o la tercera información de validez, se les asocia una información de validad correspondiente.

De este modo ya se pueden asociar las informaciones de validez, por ejemplo, por el programador de la aplicación. Esto tiene en particular la ventaja de que la duración del modo de aprendizaje para la asociación de todas las informaciones de validad necesarias se puede acortar para un escenario de aplicación.

En otras formas de realización del procedimiento, durante la ejecución de las instrucciones de programa sin información de validez se proporciona una información de señalización.

De este modo se puede reconocer preferentemente si se ejecutan instrucciones de programa que no son necesarias realmente para el escenario de aplicación. La información de señalización se puede transmitir, por ejemplo, a una consola de control o un sistema de supervisión de seguridad, de modo que los técnicos verifican, por ejemplo, en una central eléctrica, la aplicación y el equipo en el que está instalada la aplicación. De este modo es posible preferentemente reconocer las manipulaciones de la aplicación o del equipo o del entorno del equipo. Esta información de señalización se puede generar, por ejemplo, mediante una interrupción o una excepción, que está depositada en las instrucciones de programa sin información de validez.

En otras formas de realización del procedimiento, durante el encendido del modo de trabajo se retiran las instrucciones de programa sin información de validez de la aplicación.

De este modo se eleva la seguridad de la aplicación, dado que las instrucciones de programa o también los códigos de programa innecesarias se retiran de la aplicación. De este modo un atacante no tiene una posibilidad de usar las partes de programa que no se usan para el escenario de aplicación.

Según otro aspecto, la invención se refiere a un entorno de realización para la ejecución asegurada y asistida por ordenador de instrucciones de programa de una aplicación. El entorno de ejecución comprende un primer módulo de conmutación para el encendido de un modo de aprendizaje del entorno de ejecución. El entorno de ejecución comprende adicionalmente un módulo de ejecución para la ejecución de la aplicación en el entorno de ejecución mientras que el modo de aprendizaje está encendido, en donde las instrucciones de programa de la aplicación se realizan para un escenario de aplicación seleccionado predeterminado y el entorno de ejecución le asocia una primera información de validez específica al escenario de la aplicación a las instrucciones de programa realizadas. El entorno de ejecución comprende adicionalmente un segundo módulo de conmutación para el encendido de un modo de trabajo del entorno de ejecución, en donde en el modo de trabajo el entorno de ejecución verifica la primera información de validez de las instrucciones de programa, y en donde el entorno de ejecución ejecuta las instrucciones de programa en función de su información de validez.

El primer módulo de conmutación y el segundo módulo de conmutación pueden estar configurados, por ejemplo, como módulo de conmutación integral, que permite encender respectivamente el modo de aprendizaje o el modo de trabajo.

En una primera forma de realización del entorno de ejecución, el entorno de ejecución es un procesador o una máquina virtual, un núcleo de sistema operativo o un núcleo de sistema operativo usando una unidad de gestión de memoria.

Según otro aspecto, la invención se refiere a un sistema que presenta un entorno de ejecución según la invención.

Además se reivindica un producto de programa informático con instrucciones de programa para la realización del procedimiento mencionado según la invención.

Adicionalmente se reivindica una variante del producto de programa informático con instrucciones de programa para la configuración de un equipo de creación, por ejemplo, una impresora 3D o un equipo similar, en donde el equipo de creación se configura con las instrucciones de programa, de manera que se crea el entorno de ejecución según la invención.

Además se reivindica un dispositivo de facilitación para el almacenamiento y/o facilitación del producto de programa informático. El dispositivo de facilitación es, por ejemplo, un soporte de datos que almacena y/o facilita el producto de programa informático. Alternativamente y/o adicionalmente el dispositivo de facilitación es, por ejemplo, un servicio de red, un sistema informático, un sistema de servidor, en particular un sistema informático distribuido, un sistema de cálculo basado en la nube y/o sistema de cálculo virtual, que almacena y/o facilita el producto de programa informático preferentemente en forma de un flujo de datos.

Esta facilitación se realiza, por ejemplo, como descarga en forma de un bloque de datos de programa y/o bloque de datos de instrucción, preferentemente como fichero, en particular como fichero de descarga, o como flujo de datos, en particular como flujo de datos de descarga, del producto de programa informático completo. Pero, esta facilitación también se puede realizar, por ejemplo, como descarga parcial, que se compone de varias partes y en particular se descarga a través de una red Peer-to-Peer o se facilita como flujo de datos. Un producto de programa informático semejante se lee, por ejemplo, usando el dispositivo de facilitación en forma del soporte de datos en un sistema y ejecuta las instrucciones de programa, de modo que el procedimiento según la invención se lleva a un ordenador para la ejecución o configura el equipo de creación de manera que éste crea el entorno de ejecución según la invención.

Las propiedades, características y ventajas descritas anteriormente de esta invención, así como el modo y manera en cómo se consiguen se harán comprensibles de forma más clara y obvia en relación con la siguiente descripción de ejemplos de realización que se explican más en detalle en relación con las figuras. A este respecto muestran en representación esquemática:

Fig. 1 un diagrama de desarrollo de un primer ejemplo de realización del procedimiento dado a conocer;

Fig. 2 clarifica un segundo ejemplo de realización del procedimiento dado a conocer;

Fig. 3 clarifica un tercer ejemplo de realización del procedimiento dado a conocer;

Fig. 4 un entorno de ejecución de un cuarto ejemplo de realización; y

Fig. 5 un sistema con un entorno de ejecución.

En las figuras los elementos iguales funcionalmente se proveen de las mismas referencias, siempre y cuando no se indique lo contrario.

Las siguientes explicaciones se refieren a este respecto a todos los ejemplos de realización.

La fig. 1 muestra un diagrama de desarrollo de un primer ejemplo de realización del procedimiento dado a conocer 100. El procedimiento 100 es apropiado para la ejecución asegurada de instrucciones de programa de una aplicación en un ordenador o un equipo, por ejemplo un equipo de control para un sistema de extinción de incendios, un equipo de supervisión para la supervisión de picos de tensión en alta tensión o un equipo de campo.

El procedimiento 100 presenta para ello una primera etapa del procedimiento para el encendido 110 de un modo de aprendizaje de un entorno de ejecución. El entorno de ejecución sirve en particular para la ejecución de la aplicación y a este respecto se puede tratar de un procesador, un conector del sistema operativo o un procesador virtual o una máquina virtual, como por ejemplo una máquina virtual de Java. Un encendido del modo de aprendizaje está protegido, por ejemplo, con un mecanismo de seguridad, como una entrada de contraseña o autentificación de usuario protegida criptográficamente.

El procedimiento 100 comprende adicionalmente una segunda etapa del procedimiento para la ejecución 120 de la aplicación en el entorno de ejecución mientras que el modo de aprendizaje está encendido, en donde las instrucciones de programa de la aplicación se realizan para un escenario de aplicación seleccionado predeterminado y el entorno de ejecución le asocia una primera información de validez específica al escenario de la aplicación a las instrucciones de programa realizadas. Para ello se puede usar, por ejemplo, una estructura de datos, que almacena una información de validez para cada instrucción de programa realizada. A las instrucciones de programa no realizadas se les puede asociar, por ejemplo, una información de invalidez, en particular en forma de un bloque de datos puesto a cero, una excepción (inglés, exception) o una interrupción (inglés, interrupt). La estructura de datos puede contener en primer lugar como valor inicial sólo informaciones de invalidez y éstas se sobrescriben luego durante la fase de aprendizaje parcialmente o totalmente con informaciones de validez.

En una variante se determina la información de invalidez porque no está presente una información de validez de una instrucción de programa.

El procedimiento 100 comprende adicionalmente una tercera etapa del procedimiento para el encendido 130 de un modo de trabajo del entorno de ejecución, en donde en el modo de trabajo el entorno de ejecución verifica la primera información de validez de las instrucciones de programa, y en donde el entorno de ejecución ejecuta las instrucciones de programa en función de su información de validez. Si durante el modo de trabajo se debe ejecutar por tanto equivocadamente una instrucción de programa con una información de invalidez, por ejemplo, se puede realizar una excepción asociada y en particular informarse preferentemente a un administrador con una información de señalización. Pero gracias a la información de señalización, la aplicación, el entorno de ejecución o el equipo en el que se ejecuta la aplicación, también se puede llevar a un estado asegurado. Entonces no se ejecuta preferentemente la instrucción de programa correspondiente con la información de invalidez .

Un encendido del modo de trabajo está protegido, por ejemplo, con un mecanismo de seguridad, como una entrada de contraseña o una autentificación de usuario protegida criptográficamente.

Mediante el procedimiento 100 dado a conocer es posible hacer inalcanzables funcionalidades desaprovechadas de una aplicación, por ejemplo, de una biblioteca o de una aplicación durante el funcionamiento operativo.

La identificación de instrucciones de programa o partes de código desaprovechadas se realiza a este efecto de forma lo más automática y transparente posible para el usuario. Para ello se pueden proporcionar, por ejemplo, evaluaciones que indican a que instrucciones de programas se les han asociado informaciones de validez, y a cuáles se les han asociado informaciones de invalidez.

Por ejemplo, un desarrollador puede asignar según el dictamen de la evaluación todavía manualmente informaciones de validez a instrucciones de programa individuales, si esto es necesario en el marco de determinados datos de entrada para la aplicación.

En otras palabras, el entorno de ejecución protocoliza en el modo de aprendizaje las instrucciones de programa de todas las rutas de ejecución recorridas / alcanzadas. Esto está ilustrado, por ejemplo, en un segundo ejemplo de realización en la fig. 2. A este respecto, la fig. 2 muestra un primer bloque de instrucciones de programa 210, a las que está asociada respectivamente una información de validez E, un segundo bloque de instrucciones de programa 215, a las que está asociada respectivamente una información de invalidez NE, y un tercer bloque de instrucciones de programa 220, a las que está asociada respectivamente una información de validez E.

Durante el modo de aprendizaje se deben recorrer preferentemente todos los estados válidos, es decir, todos los estados que puede adoptar la aplicación para el escenario de aplicación seleccionado predeterminado.

El modo de aprendizaje se puede realizar, por ejemplo, en diferentes instantes: se puede realizar, por ejemplo, una fase de aprendizaje para un escenario de aplicación, por ejemplo, en una validación de software, en la recepción / certificación para una instalación de automatización. Pero esto también se puede realizar durante el sellado de ajustes de configuración del equipo, durante la carga o instalación del código de programa para la aplicación. Esta información se deposita junto con el código de programa ejecutado de la aplicación. Los datos 225, es decir, la información de validez E y/o la información de invalidez NE, se pueden depositar directamente en el código de programa, por ejemplo como indicador en el código de operación, como indicador en puntos de entrada (funciones) de una biblioteca de programas, que también se designan como anotaciones de código de objeto o anotación de código binario. Alternativamente estos datos se pueden depositar como metadatos separados.

En otra variante se modifica el código. Así se pueden sustituir p. ej. códigos de operación no marcados, es decir, instrucciones de programa 215 sin información de validez, por NOPs, TRAPs o excepciones.

Pero también se pueden usar varias fases de aprendizaje para la asociación de las informaciones de validez 225: Para ello se puede encender el modo de aprendizaje durante una primera fase de aprendizaje ya durante el test de programa del fabricante de un equipo, en el que se realizan test de escenarios de aplicación determinados o típicos. De este modo se puede generar un juego de datos con una configuración base con informaciones de validez para la aplicación.

Esta configuración base generada de antemano se puede seguir configurando luego en el usuario mediante un nuevo encendido del modo de aprendizaje. De este modo se puede acortar una segunda fase de aprendizaje directamente en el lugar de uso, es decir, el escenario de aplicación seleccionado predeterminado, del equipo.

Esta configuración base se puede crear, por ejemplo, a través de los datos del test del sistema, que todavía se puede afinar durante una fase de aprendizaje en el lugar de uso del equipo. Los datos aprendidos 225 (metadatos separados o código de objeto / código binario anotado), es decir, informaciones de validez y/o información de invalidez, se le pueden proporcionar en una variante a otro equipo en forma protegida criptográficamente, por ejemplo, como juego de datos firmado digitalmente.

El modo de aprendizaje también se puede encender en diferentes equipos o ejecutarse las fases de aprendizaje en diferentes equipos:

- en el mismo equipo objetivo, es decir, el equipo en el que se debe usar el escenario de aplicación seleccionado predeterminado;

- en un equipo de test igual constructivamente (p. ej. no operativo durante el uso para el escenario de aplicación seleccionado predeterminado);

- en un entorno de simulación del equipo objetivo (digital twin);

Con interpretación estricta, con el modo de aprendizaje encendido se realizan realmente todas las partes de programa asociadas con un escenario de aplicación (es decir, las instrucciones de programa de las partes de programa), que deben ser ejecutables posteriormente.

Pero en otra variante también se puede presentar un aprendizaje impreciso, es decir, en el modo de aprendizaje encendido se le asocia respectivamente una información de validez a una instrucción de programa en una zona ensanchada.

Para ello, por ejemplo, son concebibles diferentes escalonamientos: en el escalonamiento "fino" se le asocia a cada instrucción usada de una instrucción de programa, es decir, también las instrucciones de las subrutinas y bibliotecas de programas, una información de validez E.

En un escalonado "medio" se les asocian a subrutinas de las instrucciones de programa realizadas una información de validez E. Aquí se puede tener en cuenta, por ejemplo, una profundidad de llamada de subrutinas, para mantener bajo el coste para una asociación de la información de validez E en particular en el caso de funciones anidadas.

En el escalonamiento "burdo" se les asocian a las bibliotecas de programas usadas por la aplicación respectivamente una información de validez E, para asociar de este modo en particular a todas las instrucciones de programa de la biblioteca una información de validez.

La granularidad de los escalonamientos se podría controlar, por ejemplo, también en el caso del desarrollo de código de la aplicación mediante datos de estructura apropiados. Para ello un desarrollador les podría asociar manualmente una información de validez a instrucciones de programa determinadas e instrucciones de programa dependientes de ellas, instrucciones, subrutinas o bibliotecas de programas.

Esto también se puede implementar porque se tiene en cuenta las informaciones de validez, de manera que se pueden tener en cuenta las dependencias. Si, por ejemplo, es válida una ruta de instrucción (a ésta se le ha asignado como una información de validez), entonces también es válida automáticamente una segunda ruta de instrucción dependiente. A este respecto se le puede asignar a la segunda ruta de instrucción automáticamente igualmente una información de validez o la segunda ruta de instrucción es válida, por ejemplo, debido a su dependencia. En otras palabras, por ejemplo, la información de validez se puede legar de la primera ruta de instrucción a la segunda ruta de instrucción debido a la dependencia de la segunda ruta de instrucción respecto a la primera ruta de instrucción.

En el modo de trabajo sólo se aceptan o ejecutan por el entorno de ejecución preferentemente las instrucciones de programa a las que está asociada una información de validez E, lo que también se puede designar instrucción de programa marcada como permitida. Si se arranca una instrucción de programa NE sin información de validez, es decir, una instrucción de programa marcada como no permitida, esto conduce por ejemplo a una excepción o a otra instrucción que se ha depositado. O expresado en otras palabras, la instrucción que ha sobrescrito por ejemplo la instrucción de programa no permitida.

Si, después de la asociación de las informaciones de validez 225, un equipo se necesitase un juego nuevo o ampliado de las instrucciones de programa y/o funciones y/o subrutinas y/o bibliotecas mediante una reconfiguración o aplicación en un escenario de aplicación ligeramente modificado, existe la posibilidad de retirar de nuevo las informaciones de validez 225 asociadas, de forma controlada por un disparador, como por ejemplo una fecha de configuración y/o un disparador manual en una nueva fase de aprendizaje. Para ello se enciende de nuevo el modo de aprendizaje y se le asocian informaciones de validez conforme al escenario de la aplicación modificado o la reconfiguración a las instrucciones de programa. De este modo las instrucciones de programa no permitidas anteriormente (o instrucciones de programa sin información de validez) se pueden aprovechar de nuevo. Si zonas de código se hubiesen retirado por NOPs, Traps (trampa) o excepciones, estas fracciones de código se pueden restaurar a partir de una copia de seguridad de la aplicación no modificada o sus bibliotecas de programas.

En otra variante se implementa el procedimiento directamente en el hardware. En este caso un procesador, que representa el entorno de trabajo, por ejemplo una CPU, dispone de un primer modo, el modo de aprendizaje, y un segundo modo, el modo de trabajo. En el primer modo se marcan los códigos de operación realizados, es decir, las instrucciones de programa realizadas, es decir, se asocia una información de validez, y deposita en la imagen del

programa. Después de la conmutación al modo de trabajo sólo se aceptan todavía las instrucciones marcadas de una aplicación.

En otra variante, el procedimiento se implementa mediante un entorno de trabajo en forma de una máquina virtual. La máquina virtual registra en un modo de aprendizaje las líneas de código terminadas de una aplicación, es decir, las instrucciones de programa realizadas de la aplicación, y genera un fichero de protocolo que contiene las informaciones de validez asociadas a las instrucciones de programa. La asociación de la información de validez se puede realizar directamente durante la generación de una entrada en el fichero de protocolo.

Durante una ejecución posterior de la aplicación, la máquina virtual examina en un modo de trabajo usando el fichero de protocolo si se debe ejecutar una instrucción de programa. La integridad / autenticidad del fichero de protocolo puede estar asegurada, por ejemplo, mediante una firma digital, que se verifica por la máquina virtual.

La fig. 3 clarifica un tercer ejemplo de realización del procedimiento dado a conocer. En este ejemplo de realización se implementa el procedimiento mediante un núcleo de sistema operativo. La aplicación se inicia en el modo de aprendizaje del núcleo de sistema operativo en un entorno controlado, por ejemplo un depurador. Se detectan las instrucciones de programa 310, 320 realizadas, en tanto que a éstas se les asocia una información de validez. Las instrucciones de programa y la información de validez se pueden almacenar entonces. En el modo de trabajo, el cargador de tiempo de ejecución usa estas informaciones de validez para instrumentar el código de programa, es decir, las instrucciones de programa de la aplicación.

Por ejemplo, todas las líneas de código no realizadas, es decir, las instrucciones de programa sin información de validez, se pueden sustituir por Traps.

En otra variante el procedimiento se implementa mediante un núcleo de sistema operativo usando la unidad de gestión de memoria, también denominada Memory Managment Unit, MMU.

Una aplicación se inicia en el modo de aprendizaje del núcleo de sistema operativo en un entorno controlado, por ejemplo un depurador. Se detectan las instrucciones de programa realizadas, en las que a éstas se les asocia respectivamente una información de validez. Las instrucciones de programa y la información de validez se pueden almacenar entonces.

A las instrucciones de programa, a las que no se les asocian una información de validez, es decir, informaciones de programa sin información de validez - también denominadas partes de código no marcadas, se caracterizan por la unidad de gestión de memoria como no ejecutables o como no legibles. En el caso de acceso (carga de las instrucciones de programa de memorias externas en caché) se desencadena preferentemente por parte de la unidad de gestión de memoria una trampa o una interrupción (inglés, interrupt).

La fig. 4 muestra un entorno de ejecución 400 de un cuarto ejemplo de ejecución para la ejecución asegurada y asistida por ordenador de las instrucciones de programa de una aplicación. El entorno de ejecución 400 comprende un primer módulo de conmutación 410, un módulo de ejecución 420, un segundo módulo de conmutación 430 y una interfaz 485, que están conectadas entre sí en comunicación a través de un bus 480.

El primer módulo de conmutación 410 enciende un módulo de aprendizaje del entorno de ejecución 400.

El módulo de ejecución 420 ejecuta la aplicación en el entorno de ejecución 400, mientras que el modo de aprendizaje está encendido, en donde las instrucciones de programa de la aplicación se realizan para un escenario de aplicación seleccionado predeterminado y el entorno de ejecución 400 le asocia una primera información de validez específica al escenario de la aplicación a las instrucciones de programa realizadas.

El segundo módulo de conmutación 430 enciende un modo de trabajo del entorno de ejecución 400, en donde en el modo de trabajo el entorno de ejecución 400 verifica la primera información de validez de las instrucciones de programa, y en donde el entorno de ejecución ejecuta las instrucciones de programa en función de su información de validez.

El entorno de ejecución puede estar construido, por ejemplo, como procesador o como máquina virtual en un chip, en particular en forma de Java incorporado, en un equipo, por ejemplo, un equipo de campo, un equipo de control o un equipo de medición. El equipo puede ser parte de un sistema, en donde el equipo está conectado con una estación de trabajo de un operador a través de un bus de datos.

Un sistema semejante se muestra, por ejemplo, en la fig. 5 como quinto ejemplo de realización. En detalle la fig. 5 muestra un sistema, por ejemplo, un sistema de supervisión para alta tensión en una central eléctrica.

Una estación de trabajo, por ejemplo, un sistema informático compatible con IBM, que comprende un equipo de visualización 532, por ejemplo una pantalla, y varios equipos de entrada, por ejemplo un ratón de ordenador 533 y un

teclado 530, está conectado en comunicación con el equipo 501 a través de un tercer bus 580 y una segunda interfaz 575 de un equipo 501. El tercer bus 580 puede ser, por ejemplo, un bus de ethernet o un bus serie universal (USB).

El equipo 501 comprende un entorno de ejecución 400 para la ejecución asegurada y asistida por ordenador de instrucciones de programa de una aplicación en un escenario de aplicación seleccionado predeterminado, por ejemplo, una supervisión de alta tensión con los procedimientos de medición correspondientes. En el equipo 501 el entorno de ejecución está conectado, por ejemplo, a través de un segundo bus 585 mediante la interfaz 485 con un equipo de detección 510, por ejemplo, un sensor medidor de tensión, y la segunda interfaz 575.

Si, por ejemplo, durante la supervisión de alta tensión o de picos de tensión se ejecuta una instrucción de programa, a la que no está asociada una información de validez, entonces el entorno de ejecución puede realizar una excepción que se le muestra entonces a un operador en la estación de trabajo. El operador puede llevar a cabo entonces eventualmente verificaciones de si se ha mantenido en un comportamiento erróneo casual de la aplicación o si ha tenido lugar una manipulación de la aplicación por un tercero no autorizado.

**REIVINDICACIONES**

**1.** Procedimiento (100) para la ejecución asegurada y asistida por ordenador de una aplicación con las siguientes etapas del procedimiento:

- encendido (110) de un modo de aprendizaje de un entorno de ejecución (400);

- ejecución (120) de la aplicación en el entorno de ejecución (400) mientras que el modo de aprendizaje está encendido, en donde

- las instrucciones de programa de la aplicación se realizan para un escenario de aplicación seleccionado predeterminado;

- el entorno de programa (400) les asocia a las instrucciones de programa realizadas (210, 220, 310, 320) una primera información de validez específica al escenario de la aplicación;

- encendido (130) de un modo de trabajo del entorno de ejecución (400), en donde en el modo de trabajo el entorno de ejecución (400) verifica la primera información de validez de las instrucciones de programa, y en donde el entorno de ejecución (400) ejecuta las instrucciones de programa en función de su información de validez.

**2.** Procedimiento (100) según la reivindicación 1, en donde las instrucciones de programa realizadas (210, 220, 310, 320) se asocian a rutas de ejecución recorridas de la aplicación y una segunda información de validez específica al escenario de la aplicación se asocia respectivamente a una ruta de ejecución.

**3.** Procedimiento (100) según la reivindicación 1 o 2, en donde

- la primera información de validez se asocia durante una primera fase de aprendizaje;

- durante una segunda fase de aprendizaje el entorno de programa (400) les asocia a las instrucciones de programa realizadas (210, 220, 310, 320) una tercera información de validez específica al escenario de la aplicación.

**4.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde como entorno de ejecución (400) se usa un procesador y/o una máquina virtual y/o un núcleo de sistema operativo o un núcleo de sistema operativo usando una unidad de gestión de memoria.

**5.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde mediante un disparador se borran la primera información de validez y/o la segunda información de validez y/o la tercera información de validez de las instrucciones de programa.

**6.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde la primera información de validez y/o la segunda información de validez y/o la tercera información de validez se almacenan de forma protegida por seguridad.

**7.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde el encendido del modo de aprendizaje está protegido mediante un mecanismo de seguridad.

**8.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde la primera información de validez y/o la segunda información de validez y/o la tercera información de validez se le proporcionan a otro equipo.

**9.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde la ejecución (120) de la aplicación en el entorno de ejecución (400) con el modo de aprendizaje encendido se realiza en un equipo y/o en un equipo de test igual constructivamente y/o un entorno de simulación del equipo.

**10.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde la primera información de validez y/o la segunda información de validez y/o la tercera información de validez se asocian a las instrucciones de programa bajo la forma de instrucciones y/o de subrutinas y/o de bibliotecas.

**11.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde las instrucciones de programa, que dependen de las instrucciones de programa con la primera información de validez y/o la segunda información de validez y/o la tercera información de validez, se asocian a una información de validad correspondiente.

**12.** Procedimiento (100) según una de las reivindicaciones anteriores, en donde durante la ejecución de instrucciones de programa sin información de validez se proporciona una información de señalización.

**13.** Procedimiento (100) según una de las reivindicaciones 1 a 11, en donde las instrucciones de programa sin información de validez se retiran de la aplicación al encender el modo de trabajo.

**14.** Entorno de ejecución (400) para la ejecución asegurada y asistida por ordenador de instrucciones de programa de una aplicación, que presenta:

- un primer módulo de conmutación (410) para el encendido de un modo de aprendizaje del entorno de ejecución (400);

- un módulo de ejecución (420) para la ejecución de la aplicación en el entorno de ejecución (400) mientras que el modo de aprendizaje está encendido, en donde

  - las instrucciones de programa de la aplicación se realizan para un escenario de aplicación seleccionado predeterminado;

  - el entorno de programa (400) les asocia a las instrucciones de programa realizadas (210, 220, 310, 320) una primera información de validez específica al escenario de la aplicación;

- un segundo módulo de conmutación (430) para el encendido (130) de un modo de trabajo del entorno de ejecución (400), en donde en el modo de trabajo el entorno de ejecución (400) verifica la primera información de validez de las instrucciones de programa, y en donde el entorno de ejecución (400) ejecuta las instrucciones de programa en función de su información de validez.

**15.** Entorno de ejecución (400) según la reivindicación 14, en donde el entorno de ejecución (400) es un procesador o una máquina virtual o un núcleo de sistema operativo o un núcleo de sistema operativo usando una unidad de gestión de memoria.

**16.** Sistema con un entorno de ejecución (400) según una de las reivindicaciones 14-15.

**17.** Producto de programa informático con instrucciones de programa para la realización del procedimiento según una de las reivindicaciones 1-13.

**18.** Producto de programa informático con instrucciones de programa para un equipo de creación, que se configura mediante instrucciones de programa, para crear el entorno de ejecución (400) según una de las reivindicaciones 14-15.

**19.** Dispositivo de facilitación para el producto de programa informático según la reivindicación 17 o 18, en donde el dispositivo de facilitación almacena y/o proporciona el producto de programa informático.

FIG 1

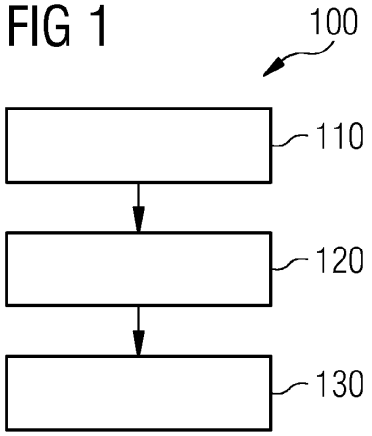



FIG 2

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 225 |
| 210 | 000000013FE38C74 | cpm | qword ptr | [rsp+0C0h], 0 | E |
| | 000000013FE38C7D | je | 13FE38C99 | | E |
| 215 | 000000013FE38C7F | cmp | dword ptr | [rsp+0C8h], 80h | NE |
| | 000000013FE38C8A | je | 13FE38CB0h | | NE |
| | 000000013FE38C8C | cmp | dword ptr | [rsp+0C8h], 0C0h | NE |
| | 000000013FE38C97 | je | 13FE38CB0h | | NE |
| 220 | 000000013FE38C99 | cpm | qword ptr | [rsp+0C8h], 100h | E |
| | 000000013FE38CA4 | je | 13FE38CB0h | | E |

FIG 3

| | | | | |
|---|---|---|---|---|
| 310 | 000000013FE38C74 | cpm | qword ptr | [rsp+0C0h], 0 |
| | 000000013FE38C7D | je | 13FE38C99 | |
| 315 | 000000013FE38C7F | int | 2A | |
| | 000000013FE38C8A | int | 2A | |
| | 000000013FE38C8C | int | 2A | |
| | 000000013FE38C97 | int | 2A | |
| 320 | 000000013FE38C99 | cpm | qword ptr | [rsp+0C8h], 100h |
| | 000000013FE38CA4 | je | 13FE38CB0h | |

FIG 4

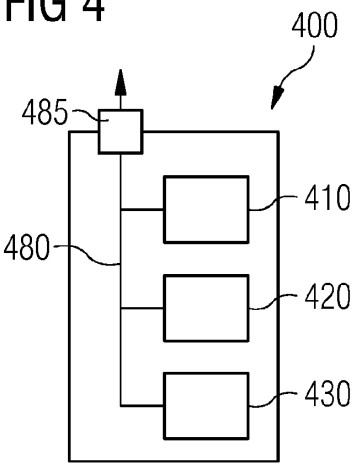
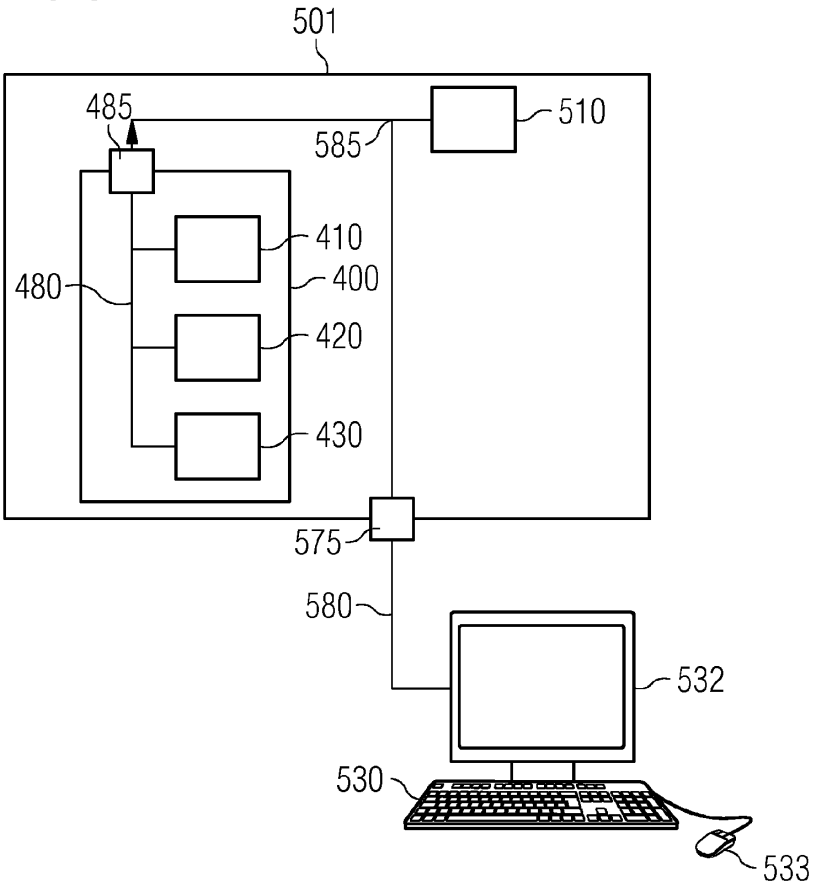

400
485
410
480
420
430

FIG 5


501
485
510
585
410
400
480
420
430
575
580
532
530
533