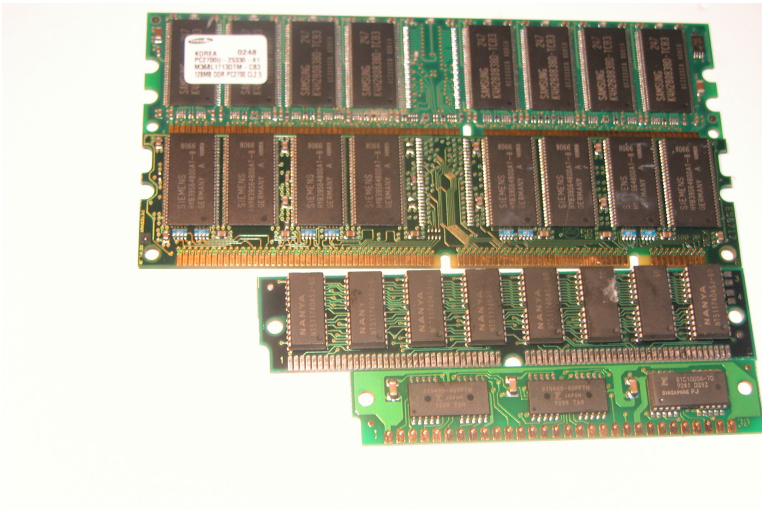
La memoria del ordenador, a fondo

Publicado el El Informatico - 17 de enero de 2022 -



Memoria RAM (Dominio público)

Todos sabemos que una memoria RAM almacena información en el ordenador (o cualquier dispositivo que siga la arquitectura de Von Neumann como puede ser un «smartphone» o tu «smart» TV). Lo que igual ya no está tan claro es cómo funciona la memoria de un ordenador. ¿Qué es la memoria física? ¿Qué es la memoria virtual? ¿Que es una página? ¿Que es el MMU del procesador? ¿Que es una dirección?

Aunque no es un conocimiento necesario a la hora de usar cualquiera de éstos dispositivos, sí es necesario conocerlo más a fondo si queremos ser programadores de verdad (y no símples picateclas). En éste artículo te doy las respuestas que necesitas.

> NOTA: Es **IMPRESCINDIBLE** haber leido y entendido previamente los artículos «Bits y bytes. Cómo funciona la información digital» y «Sistemas de numeración usados en tecnología digital» para poder entender éste artículo.

¿Qué es una memoria?

Una memoria es un dispositivo que almacena información de alguna manera. En el

artículo «Historia de los medios de almacenamiento digitales» se detallan numerosos medios de almacenamiento que sirven como medios de almacenamiento (o memorias) para la información digital.

Entre los numerosos medios de almacenamiento que se describen en ese artículo se describe también la *memoria RAM* (dentro del apartado «*Chips de memoria*«), que en la actualidad es la memoria principal de cualquier ordenador, siguiendo la arquitectura de Von Neumann. Este tipo de memorias **se caracterizan por su alta tasa de escritura y lectura de datos, debido en parte también a que el acceso a la información no es secuencial (aleatorio, o «random»), y con la característica de que la información almacenada es volátil**. Es decir, la información se pierde con el tiempo una vez se quita la alimentación de los chips que componen la memoria.

Es bastante común por parte de la gente pensar que la información se pierde de manera inmediata, **pero ese no es el caso**. **La información se pierde gradualmente con el tiempo al quitar la alimentación**. Por eso, **si reinicias el ordenador, una buena parte de los datos podría mantenerse intacta en la memoria al reiniciar**. Si bien serán sobreescritos durante el uso del ordenador.

A los «gamers» de los 90 quizás les suene de algo la frase «stop 'n' swop». Esto era una característica eliminada de los juegos Banjo Kazooie y Banjo Tooie, donde supuestamente sería posible transferir datos entre ambos juegos cambiando los cartuchos de ambos juegos, gracias a que los datos se mantendrían en la memoria durante un corto periodo de tiempo. Esta característica al final no llegó a implementarse en Banjo Tooie debido a que era demasiado complicado realizar el cambio en un periodo de tiempo muy corto (apenas unos segundos).

¿Cómo se estructura la memoria RAM por dentro?

La memoria permite almacenar *bytes* de información dentro de una matriz (array) de bytes en orden secuencial. Aunque el orden de los bytes es secuencial, el acceso a cada byte se puede realizar de manera aleatoria dentro de la memoria (de ahí el nombre de RAM, o Random Access Memory). Esto quiere decir que puedes acceder a cualquier parte de la información en cualquier momento.

Address	Content (8 bits)
…	…
0x00000007	0x88
0x00000006	0x77
0x00000005	0x66
0x00000004	0x55
0x00000003	0x44
0x00000002	0x33
0x00000001	0x22
0x00000000	0x11

Ejemplo de array de bytes en una

memoria, representado en una tabla.

Al valor que representa la posición de cada byte de la memoria lo llamamos la **dirección**. Los valores de las direcciones de la memoria se representan en el sistema de numeración **hexadecimal**. Por ejemplo, el séptimo byte de la memoria tendría la dirección 0x00000007. Y el décimo, 0x0000000A.

Comunicación entre el procesador y la memoria

La comunicación entre el procesador y la memoria se realiza a traves de los tres principales buses de cualquier ordenador: el bus de datos (Representa la información. Esta información es bidireccional, pero sólo puede ir en un único sentido en cada momento), el bus de direcciones (sirve para especificar la dirección de memoria que estamos usando), y el bus de control (sirve, entre otras cosas, para hacer referencia a qué elemento dentro del ordenador estamos usando, y de qué manera lo estamos usando. Por ejemplo y en el caso de la memoria, si estamos leyendo, o escribiendo datos).

> NOTA: Un 'bus' es un conjunto de conductores, pistas o hilos sobre los que viaja la información. Podría ser un conjunto de pistas de cobre en el circuito, cables, etc, que interconectan los diferentes componentes del circuito digital. Los tres principales buses (Datos, dirección y control) son esenciales en cualquier sistema computacional.

En el caso de los ordenadores convencionales al menos, el procesador no está necesariamente conectado a ninguno de los componentes de forma directa. En lugar de ello, se conecta a traves de otro chip que sirve de interfaz entre los componentes en la placa base. Este chip puede ser el puente norte/sur (En las placas base antíguas), o **lo que denominamos el chipset** (en las placas base modernas), que es un conjunto de circuitos integrados en un único chip (puente norte/sur, controladores, etc).

En el caso de la memoria y el procesador, el intermediario es el **MC (Memory Controller)**. Que puede estar en el puente norte de las placas base más antíguas (Y se conecta a traves del bus FSB con el procesador, o Front Side Bus), o en el caso de los ordenadores modernos estaría integrado en el propio chip de la CPU como un elemento más (IMC, o Integrated Memory Controller), con el fin de reducir la latencia. **Este es el elemento encargado de controlar la comunicación entre la memoria física instalada en la placa base y el procesador**.

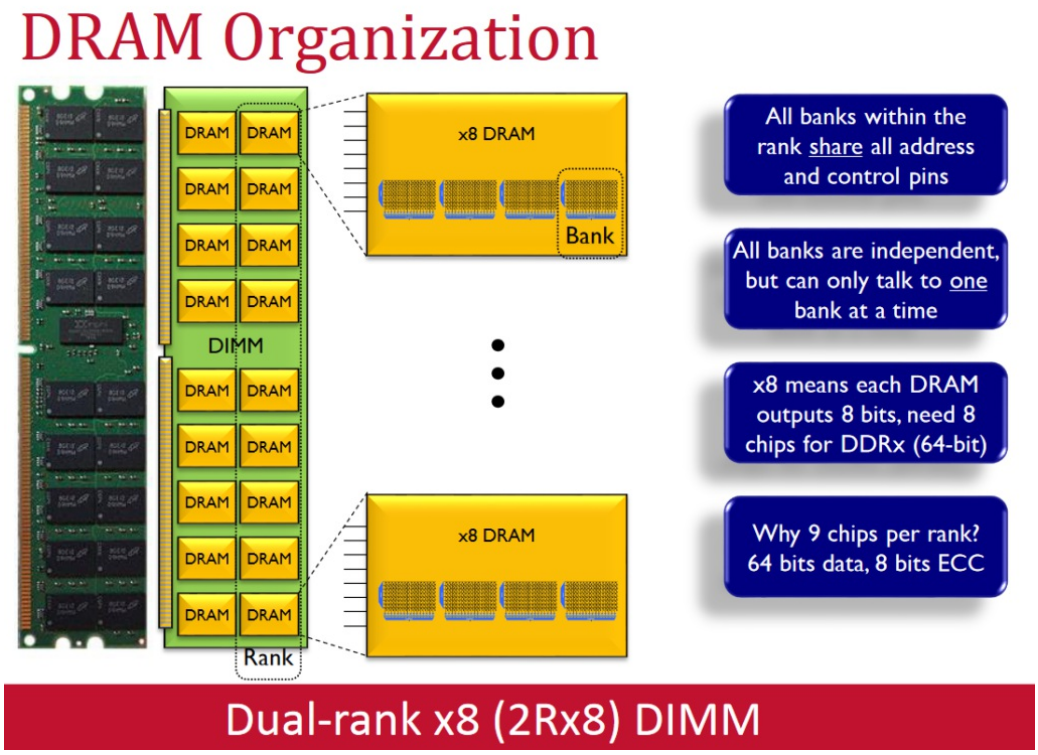
Memory Banking y Memory Rank

Explicado de manera sencilla, una única memoria RAM almacena la información como una serie consecutiva de bytes, pero en la práctica evidentemente no es tan sencillo. Si te das cuenta, la memoria RAM de un ordenador dispone de varios chips en su circuito integrado. Aparte de los controladores y otros componentes necesarios, cada uno de los chips es una memoria. Y la información se tiene que almacenar y distribuir entre todos y cada uno de los chips. ¿Cómo se gestiona eso?

Para gestionar distintas regiones de la memoria (por ejemplo, en diferentes chips) se usa una técnica denominada ***Memory Banking***, que consiste en dividir la memoria en diferentes bancos de memoria. El controlador de la memoria sólo tiene que decidir a qué chips le corresponde una determinada dirección de la memoria y cambiar de banco cuando sea necesario (bank switching).

En el caso de las memorias dinámicas (DRAM) y las basadas en ésta tecnología (DDR SDRAM), **cada uno de los chips de memoria representa un banco de datos**, que comparten la mísma dirección y el mismo bus de control. **Cada banco de memoria funciona de manera independiente, y sólo es posible comunicarse con un banco de memoria al mismo tiempo**.

La información de la memoria se distribuye en filas de chips, llamadas Memory Ranks. Cada fila está formada por 9 chips de memoria. Ocho de los chips son usados para almacenar la información de manera distribuida entre cada chip (8 bits x 8 chips = 64 bits de información), mientras que el último chip se usa como protección frente a errores (Error Correction Code o ECC).



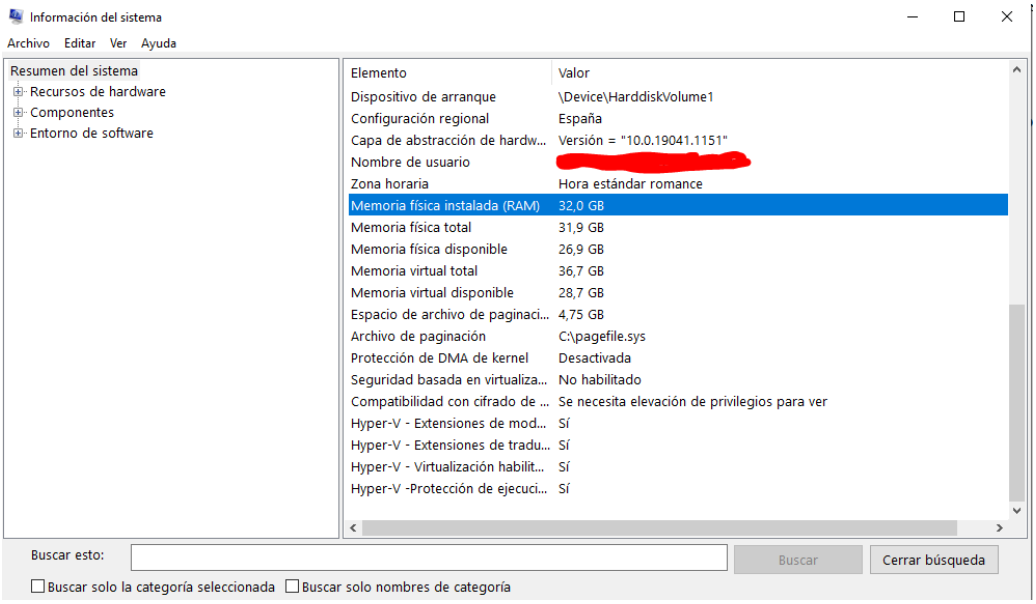
Esquema de la estructura física de una memoria RAM. Fuente: Stony Brook University

Memoria física y Memoria virtual

Es bastante común en un ordenador tener no una, sino varias unidades de memoria RAM instaladas en la placabase. Sin embargo, a la hora de iniciar el ordenador, se nos presenta la suma de la capacidad de todas y cada una de las memorias instaladas en el sistema. Por ejemplo, si tienes dos unidades de memoria RAM de 16GB, la capacidad total sería de 32GB, presentada al usuario como una única memoria.

A la memoria disponible de manera *física* se le llama *memoria física*. Y es el total (teórico) de memoria del que disponemos.

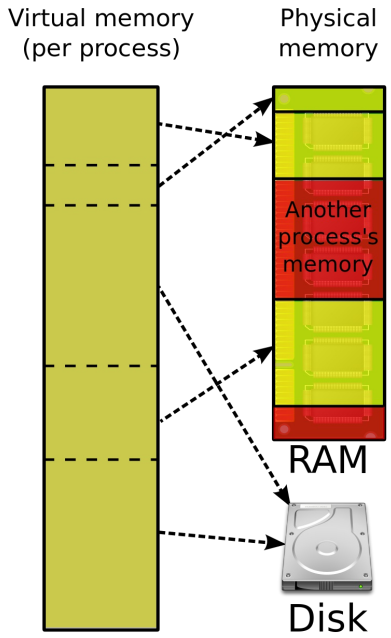
En Windows, podemos comprobar el estado de la memoria instalada desde la utilidad de información del sistema.



Pantalla de información del sistema de Windows

La información del sistema nos indica que tenemos un total de 32GB de memoria, de los cuáles hay disponibles 31,9GB (en JEDEC) para el sistema operativo, descontando lo que usa la UEFI y lo que se se reserva para otros propositos. De esos 31,9GB se descuenta lo que usa el sistema operativo en si, y nos quedan 26,9GB en total.

Este es el total de memoria disponible en la RAM (física). Sin embargo, exísten otros tipos de memoria. Por ejemplo, el archivo de paginación de Windows (o la partición SWAP en Linux). El sistema usa todas las memorias a su disposición y las junta como si todo fuese una única memoria (un espacio de direcciones de memoria contíguas). A ésta memoria se le denomina **memoria virtual**.



Representación de la memoria virtual y física (Fuente: Wikipedia. CC BY-SA)

Por éste motivo, la capacidad de memoria virtual será superior a la memoria física. En efecto, si sumas el tamaño del archivo de paginación (4,75GB) a la capacidad de memoria total física (31,9GB) te dan los 36,7GB de memoria virtual total que reporta la pantalla de información del sistema. Descontando la memoria reservada para el sistema, el sistema dispone de un total de 28,7GB de memoria para apliaciones (La memoria virtual disponible). Esta memoria virtual es la que después se usa para otorgar memoria a cada proceso del sistema.

> **NOTA**: Recuerda que las capacidades reportadas por Windows siguen el estandard JEDEC y no el ISO. Así pues, las unidades se muestran en múltiplos de 1024 (ejemplo, 1GB = 1024MB), y no de 1000 (1GB en ISO = 1000MB y 1GiB en ISO = 1024MB).

Paginación de la memoria

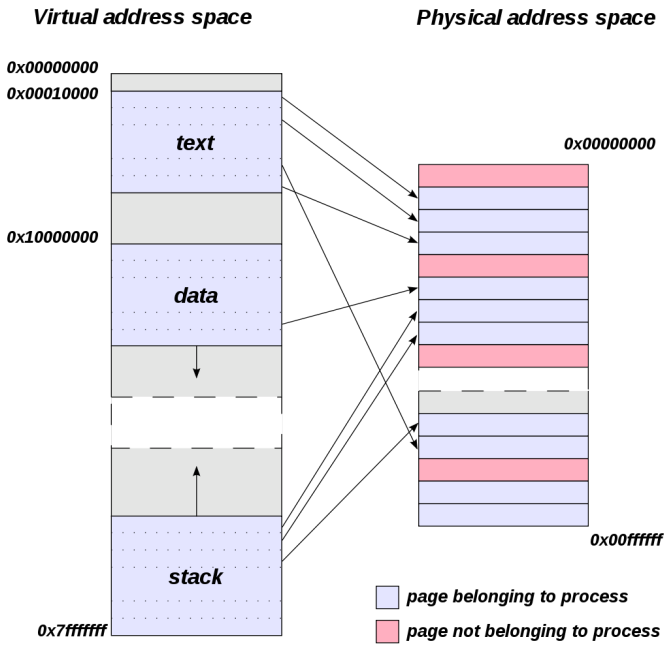
Los sistemas operativos modernos dividen la memoria virtual en pequeños pedazos de memoria de un tamaño expecífico. A esos trozos de memoria se les denomina **páginas**. Cuando una aplicación solicita más memoria al sistema, el sistema le otorgará a la aplicación tantas páginas como requiera (siempre y cuando sea posible), y de un tamaño dentro de los límites establecidos.

Por lo general, **el tamaño de las páginas no puede ser inferior a 4KiB**. Dependiendo de la arquitectura, **en x86 (32-bits) el límite superior está entre 4MiB (Sin PAE) o 2MiB (Con PAE). En x86_64 (64 bits) el limite superior está en 2MiB, pudiendo aceptar bajo algunas condiciones páginas de hasta 1GiB** (Si lo soporta la CPU). Otras arquitecturas pueden tener otros límites distintos.

Protección de la memoria

Las páginas de memoria son propiedad del proceso al que se le asignan, de modo que otro proceso necesitará pedir permiso antes a traves del sistema si quiere acceder a la memoria de otro proceso externo y ajeno al mísmo, pudiendo acceder a la memoria de los procesos con el mismo nivel de privilegios. No obstante, los procesos pueden proteger esas páginas para evitar ciertas operaciones sobre la mísma. Generalmente, se pueden proteger contra lectura y/o escritura. En los sistemas de 64-bits (o que hagan uso de las extensiones PAE del procesador), es también posible hacer uso del bit NX (No eXecute).

El bit NX previene la ejecución de código arbitrario almacenado en las páginas marcadas con ese bit. Esto supone una protección contra cierto tipo de ataques como, por ejemplo, la ejecución de código arbitrario mediante desbordamiento de buffer.



Esquema del espacio de memoria virtual otorgada a un proceso y su correspondiente uso del espacio de memoria física (Fuente: Wikipedia) (BSD)

Para ser más especifico, los sistemas Windows de 32 bits (desde Windows XP SP2 y Windows Server 2003) permiten también el uso de las extensiones PAE y el bit NX mediante la característica DEP (Data Execution Prevention). No obstante, en Windows XP (y todos los sistemas domésticos Windows de 32 bits) ésta caracteristica

sólo viene habilitada por defecto para los procesos esenciales del sistema.

Esto significa que si la característica DEP está deshabilitada para el resto, es posible ejecutar código arbitrario en cualquier página de memoria. Este no es el caso de los sistemas de 64 bits, que incorporan protección mediante hardware (desde el procesador) para evitar la ejecución de código mediante el bit NX. Con lo que la característica DEP sólo se reserva a procesos esenciales del sistema.

Otras características incluyen la aleatorización de los espacios de memoria de los ejecutables, y la separación de las diferentes secciones del ejecutable en diferentes areas de la memoria con distintos niveles de protección.

Unidad de gestión de la memoria (MMU)

Todo lo que acabo de mencionar no se gestiona mediante software. El procesador del ordenador incorpora un elemento que denominamos la *unidad de gestión de la memoria* (**MMU, o Memory Management Unit**). Este elemento es el encargado de mantener un registro con una lista de direcciones virtuales y sus correspondientes direcciones físicas (denominada **Page Table** (PT)), realizar la conversión entre direcciones virtuales y físicas, gestionar la protección de cada página de la memoria virtual, control de caché, y sirve como intermediario en el bus de la memoria.

Toda la información relacionada con las páginas de memoria se almacena en la PT. Cada entrada de la tabla es una PTE (Page Table Entry), y en ella se incluye la dirección virtual de la página, la física correspondiente, y la información sobre la página (El proceso al que pertenece, si ha sido modificado (dirty bit), las banderas de protección, espacios de memoria, si es memoria física o no, etc).

La MMU incluye una memoria caché que se denomina **TLB (Translation Lookaside Buffer)**. En ésta caché se almacenan las entradas PTE usadas más recientemente para acelerar el proceso de traducción de las direcciones más usadas. Al solicitar el acceso a una dirección de memoria virtual, el MMU primero busca en el TLB si la dirección se encuentra ya disponible. De no ser el caso, lo busca en la PT. Al finalizar, el proceso de búsqueda, el MMU le devuelve al procesador una entrada con la información requerida para acceder a esa memoria. Por ejemplo, si se trata de memoria física o no, la dirección, etcétera.

Errores en la memoria: Page Fault

La excepción (que no error) «Page Fault» se produce normalmente cuando se intenta acceder a una dirección de memoria virtual que no se encuentra mapeada a la memoria física (pero si lo esté a un archivo de paginación, por ejemplo). En principio, aunque el archivo de paginación sirve como extensión de la memoria, su uso está limitado al almacenamiento de páginas de memoria de procesos en segundo plano, que estén pausados, etcétera, con el fin de liberar memoria en la memoria física. El

procesador no accede directamente a éstas páginas de memoria.

Es el sistema operativo es el encargado de gestionar éste proceso, de la siguiente manera:

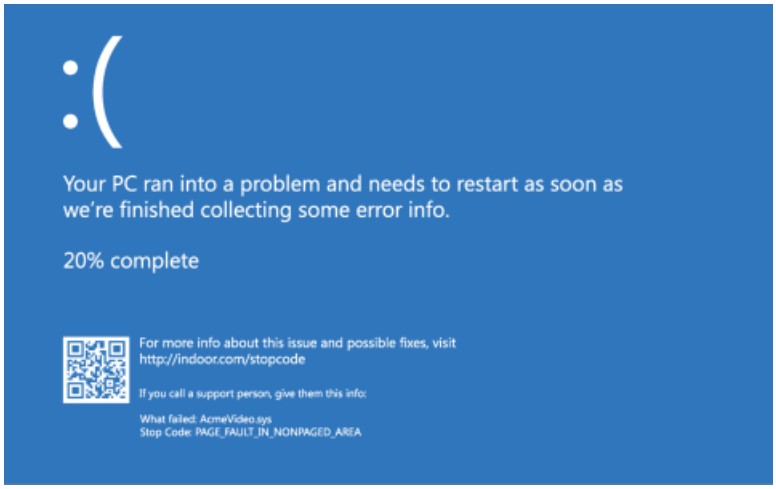
Primero comprueba que la operación sea válida. Si no lo es, entonces resulta en un error (puede ser un error de violación de acceso, o una parada del sistema dependiendo de la gravedad).

Despues, intenta encontrar una serie de páginas en la memoria física disponibles para poder cargar la página del disco duro. Está información la almacena el sistema operativo en el PCB (Process Control Block). Si encuentra páginas libres, copia las páginas del disco a la memoria física y lo mapea a la memoria virtual para poder volver a intentar el acceso a traves de éstas páginas y continuar con el proceso.

En caso de no haber páginas libres, entonces debe buscar páginas en la memoria que puedan ser temporalmente reemplazadas por las páginas en el disco. Normalmente páginas que no estén en uso o no vayan a usarse en poco tiempo.

Una vez finalizado el proceso, se actualiza la PT con la nueva información de la página y el proceso vuelve a ejecutarse desde donde estaba.

Cuando esta operación no es válida, el sistema operativo se lo hace saber al proceso mediante un error de violación de acceso, o una señal SIGSEGV. Pero si el problema lo tiene el propio nucleo del sistema, ésto puede resultar en una parada del sistema (En linux ésto sería un Kernel Panic, y en Windows un bug check o BSoD)



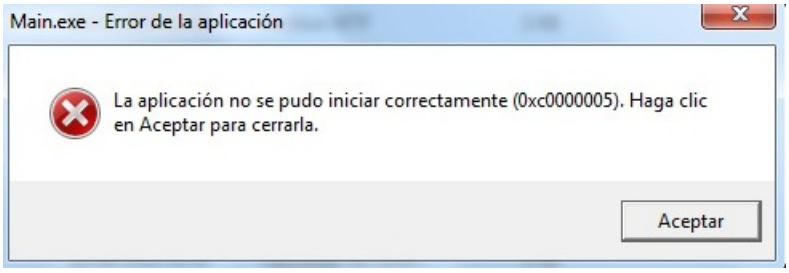
Bug check en Windows, comunmente denominado BSodD, con el código de error PAGE_FAULT_IN_NONPAGED_AREA

Cuando éste error produce un kernel panic o un bug check y se produce de manera reiterada, suele ser debido a problemas en la memoria física.

Error de violación de acceso (access violation, segmentation fault, SIGSEGV)

Este es uno de los errores más comunes a los que nos podemos enfrentar a diario. El error de violación de acceso se produce cuando un proceso intenta acceder a una dirección de memoria localizada en una página de memoria que no le corresponde al proceso, y al cuál no tiene permiso para acceder.

En Windows éste error se corresponde con el código 0xC0000005, o en Linux con la señal SIGSEGV.



Error 0xC0000005

Al ejecutar algunas rutinas, como las rutinas para abrir un archivo en el disco duro o para inicializar algun componente de hardware, la rutina normalmente devuelve un puntero de memoria. Éste puntero apunta a una dirección en concreto de memoria. Si la rutina se ejecuta correctamente, devuelve una dirección de memoria válida. No obstante, es común que cuando hay algún problema al ejecutar la rutina, el valor devuelto sea un valor neutral (normalmente 0, indicando que ha habido un error). Estos valores no se corresponden con direcciones válidas de memoria ya que están reservadas al sistema, con lo que al intentar acceder a ellas a traves del puntero se produce el error de violación de acceso.

Dada la naturaleza de éste error, puede estar causado por infinidad de problemas. Desde errores en la programación, acceso a archivos protegidos sin permiso, fallo al solicitar más memoria (que no haya más memoria), fallos en la memoria, fallos al iniciar un dispositivo de hardware (por ejemplo al intentar usar la GPU), fallo al conectar a un servicio a internet, etcétera. La mayoría de éstos fallos pueden ser detectados por el proceso antes de que se produzca éste error, ya que es posible comprobar el valor del puntero antes de acceder al mísmo. Pero eso es asumiendo que la programación sea la correcta.

ANTERIOR

Operaciones lógicas (Lógica booleana) y Bit Shifting

SIGUIENTE

Tu ISP te está atacando, aunque tú no lo sepas (o lo ignores)

Buscar ...

Entradas Recientes

- Estoy hasta las narices de la web moderna
- Ingenieria inversa básica con Ghidra
- Acerca de la nueva ley transgénero (Y sobre la disfória de género)
- Depresiones causadas por las redes sociales
- ¿Necesito saber matemáticas para aprender informática?
- ¿Es el fín de los discos duros tradicionales?

Categorías

Actualidad | Android | Básicos | Ciberseguridad | Clima | Criptografía | Electronica | Emulación / Virtualización | FOSS | Hacking | Hardware | Informática | Internet | Juegos | Opinion | Otros | Personal | Privacidad | Programación | Tecnología | Time Machine | Tutoriales

RSS

Subscribirse al feed RSS

Acerca de mi

Acerca de ElInformati.co