|  |  |  |
| --- | --- | --- |
| **Fecha de entrega:** 01 de septiembre de 2017 | **Título de la tarea:** TBL cache memory and virtual memory. | **Número de la tarea:** 5 |
| **Grupo:** 3CM3 | **Alumno:** Estrada Granados Diego | **Unidad de aprendizaje:** Arquitectura de Computadoras |

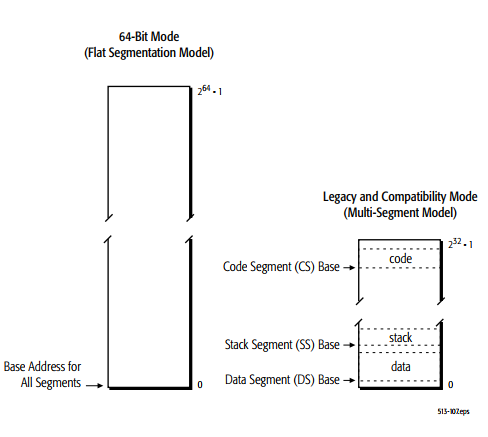
**TBL cache memory and virtual memory.**

**Desarrollo:**

La memoria virtual consiste en todo el espacio de direcciones disponible para los programas. Es una gran dirección lineal espacio que se traduce por una combinación de hardware y software del sistema operativo a una dirección física. La memoria virtual es tratada de la siguiente manera:

* Modo de 64 bits. Utiliza un modelo de segmentación plano de la memoria virtual. La memoria virtual de 64 bits el espacio se trata como un espacio de direcciones único, plano (no segmentado). El acceso a las direcciones del programa puede estar ubicado en cualquier lugar de espacio lineal de 64 bits. El sistema operativo puede utilizar separadores selectores pila y los segmentos de datos con fines de protección de la memoria, pero la dirección base de todos estos segmentos siempre es 0.
* Modo de compatibilidad: Este modo utiliza un modelo protegido de varios segmentos de memoria virtual, al igual que en el modo protegido heredado. El espacio de memoria virtual de 32 bits se trata como un conjunto segmentado de direcciones espacios para segmentos de código, pila y datos, cada uno con su propia dirección base y parámetros de protección. Se especifica un espacio segmentado añadiendo un selector de segmentos a una dirección. (Advanced Micro Devices, Inc, 2013)

En la siguiente imagen se muestra como es tratada la memoria virtual en estos casos especificados.



Los sistemas operativos han utilizado la memoria segmentada como un método para aislar los programas de los datos que utilizan, en un esfuerzo por aumentar la fiabilidad de los sistemas que ejecutan múltiples programas simultáneamente. Sin embargo, la mayoría de los sistemas operativos modernos no utilizan las funciones de segmentación disponibles en la arquitectura heredada x86. En su lugar, estos sistemas operativos manejan funciones de segmentación enteramente en software. Por esta razón, la arquitectura AMD64 dispensa la mayoría de las funciones de segmentación heredadas en el modo de 64 bits. Esto permite que los sistemas operativos de 64 bits se codifiquen de forma más sencilla y admite una gestión más eficiente de entornos multitarea que lo que es posible en la arquitectura x86 heredada. (Advanced Micro Devices,Inc , 2013)

En el modo compatibilidad, todos los registros de segmentos son accesibles para el software. Las estrategias de gestión de memoria traducen las direcciones generadas por los programas en direcciones en la memoria física mediante segmentación y / o paginación. La gestión de memoria no es visible para los programas de aplicación. Es manejado por el sistema operativo y el hardware del procesador. Los *Translation-lookaside buffers (TLBs),* también conocidos como cachés de traducción de páginas, eliminan casi la penalización de rendimiento asociada con la traducción de páginas. Los TLB son cachés especiales en el chip que contienen las traducciones de direcciones virtuales más utilizadas recientemente a las direcciones físicas. Cada referencia de memoria (instrucciones y datos) es verificada por el TLB. Si la traducción está presente en el TLB, se proporciona inmediatamente al procesador, evitando así las referencias de memoria externa para acceder a las tablas de páginas. Los TLBs aprovechan el principio de la localidad. Es decir, si se hace referencia a una dirección de memoria, es probable que las direcciones de memoria cercanas se hagan referencia en un futuro próximo. (Advanced Micro Devices, Inc, 2013)

# Trabajos citados

Advanced Micro Devices, Inc. (Mayo de 2013). *AMD Developer Cetral.* Obtenido de AMD64 Architecture Programmer's Manual Volume 2: System Programming: http://developer.amd.com/wordpress/media/2012/10/24593\_APM\_v21.pdf

Advanced Micro Devices,Inc . (Octrubre de 2013). *AMD Developer Central .* Obtenido de AMD64 Architecture Programmer's Manual Volume 1: Application Programming: https://support.amd.com/TechDocs/24592.pdf