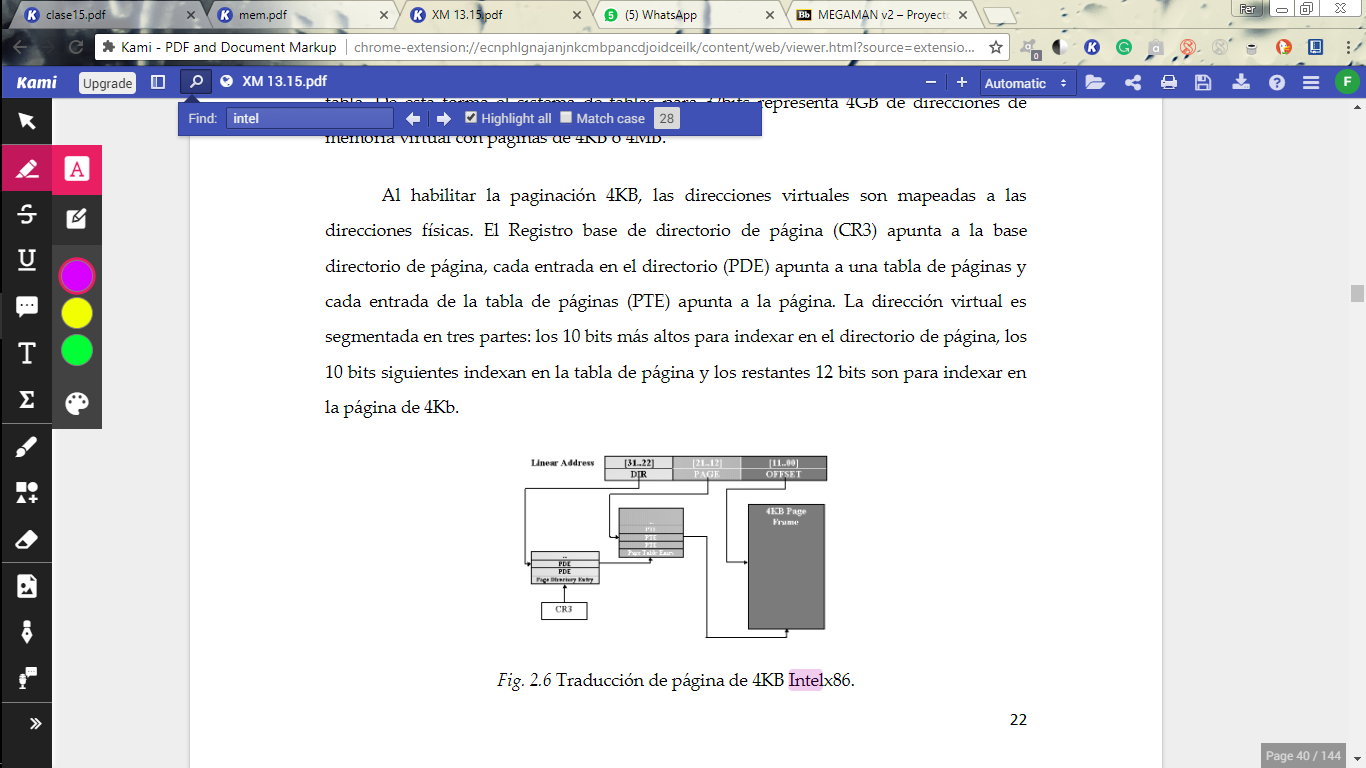
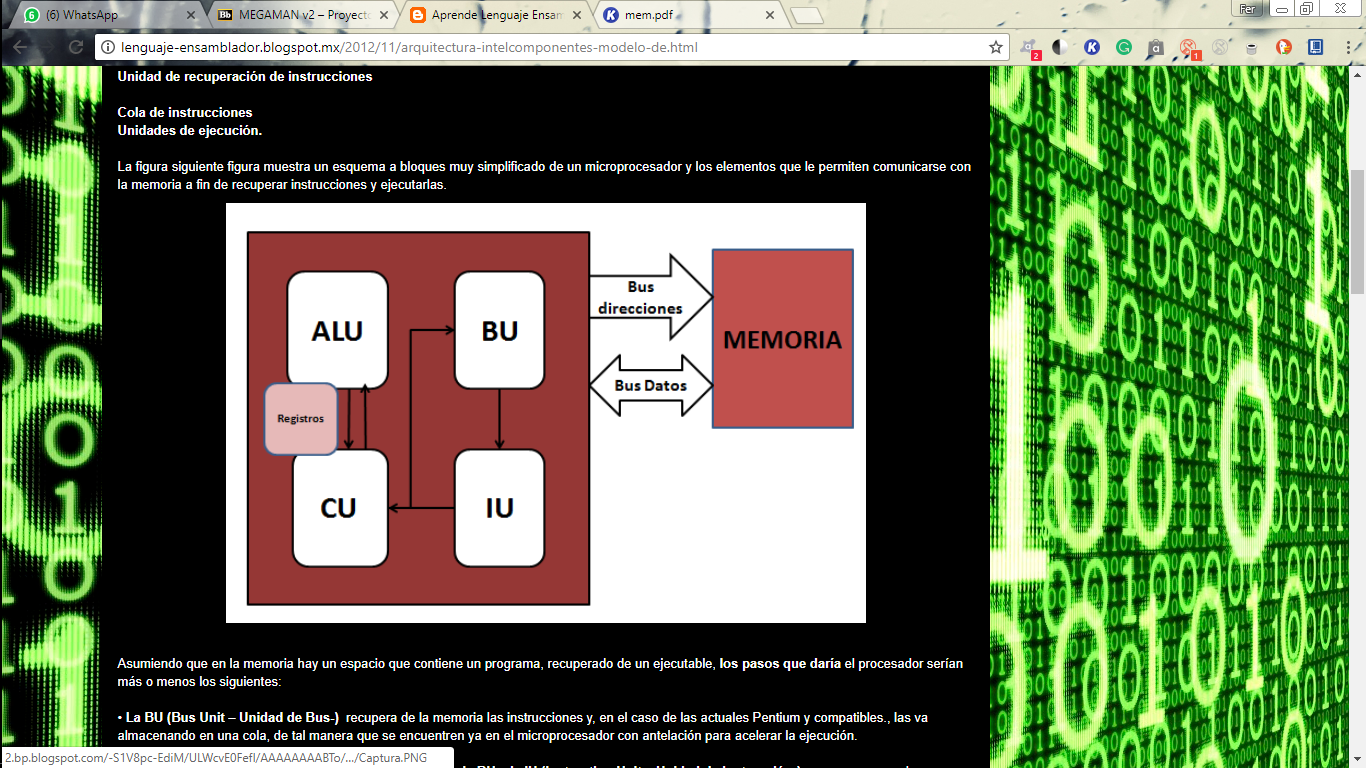
**Manejo de memoria**

***En clase vimos varios métodos de manejo de memoria y mencionamos que para el manejo de memoria el CPU debe proveer cierto apoyo, en una unidad de manejo de memoria (MMU por sus siglas en ingles).***

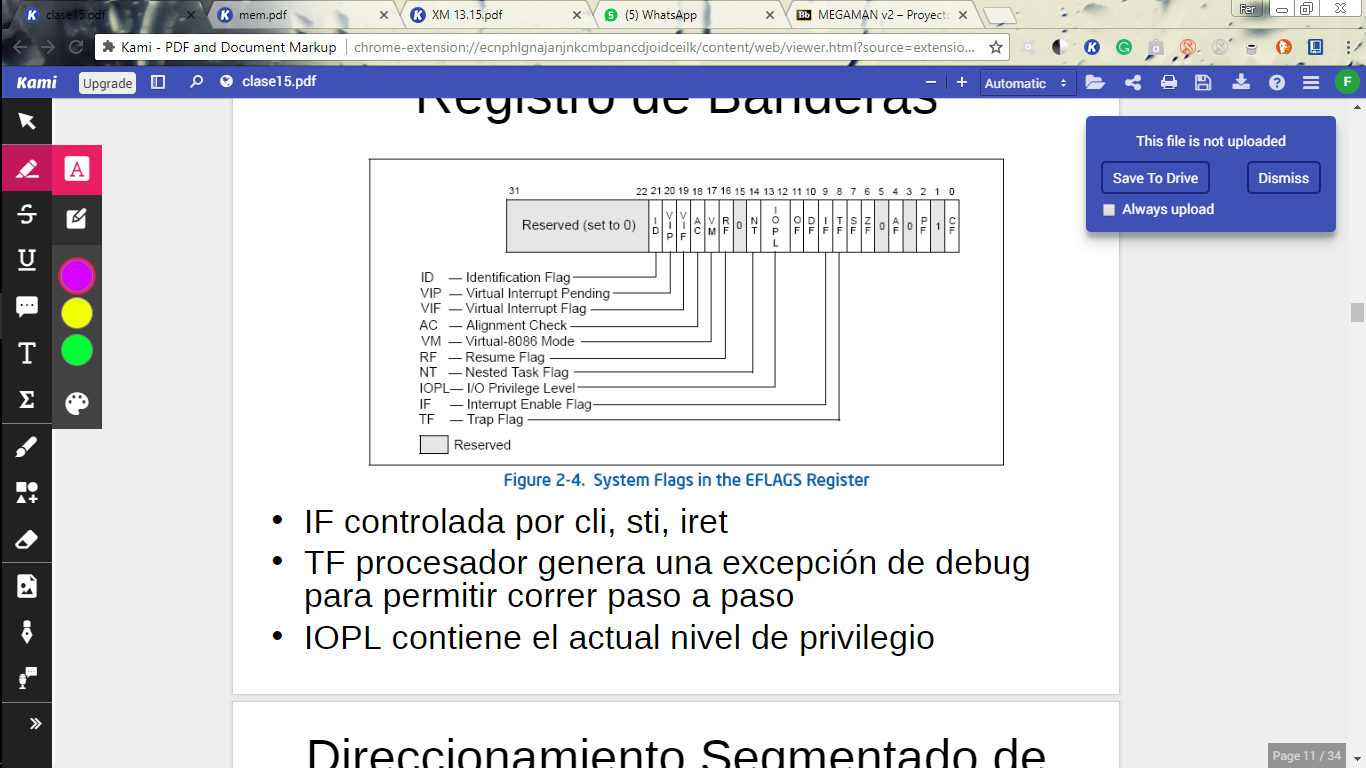
**Problema 1**

Los procesadores de Intel se ejecutan de uno de tres modos: modo real, modo protegido o modo SMM. En teoría, podría haber un cuarto modo pero se considera un submodo dentro del modo real. Al arrancar el equipo comienza en el modo real, sin embargo, el modo nativo del procesador es el modo protegido. Una de las ventajas de los procesadores Intel es que cuentan con una memoria virtual, que permite ejecutar programas mucho más grandes que la cantidad disponible de la memoria del ordenador. Esto se logra mediante la unidad de manejo de memoria que crea mapas de los segmentos llamados marcos de página por medio de tablas, una para el directorio de paginación y otra para la tabla de páginas. En el directorio de página, cada entrada hace referencia a una tabla de página, y en cada una de las entradas de la tabla de página apunta hacia una dirección física que es asignada a la dirección virtual. La manera de obtener la dirección física es por medio de la segmentación en grupo de bits de la dirección virtual, y con ella se realiza el desplazamiento dentro del directorio y el desplazamiento dentro de la tabla.

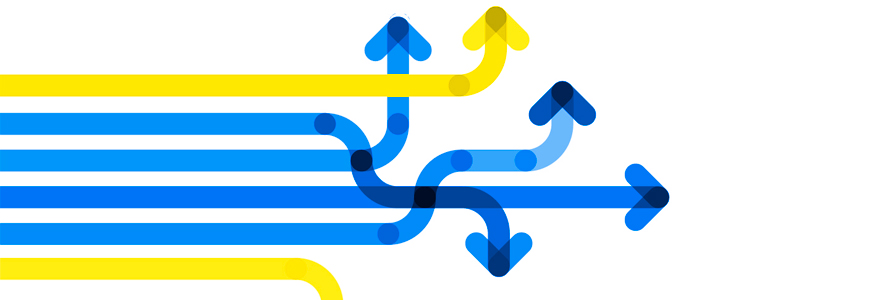
Los procesadores Intel, organizan la memoria como una secuencia de bytes, direccionable mediante su *bus de address*. La memoria que está conectada a este bus, se le llama memoria física y está lleno de direcciones físicas.

La parte de BU, unidad de bus, es la encargada de recuperar la memoria de las instrucciones, y puede ser el caso de que también se use una cola donde se van almacenando. Después, las instrucciones pasan a la IU, unidad de instrucción, que se encarga de decodificar las instrucciones y recuperar los datos que pudieran llevar. Posteriormente, pasan a la CU, unidad de control, y al ALU, unidad lógica aritmética, que es donde ambas partes comparten la parte de los registros.

Los procesadores Intel cuentan con cuatro diferentes tipos de registro. El registro de datos, que cuenta con 16 bits, dividido en dos con 8 bits cada uno, posee algunas banderas del CR0, habilita la paginación de la memoria, habilita y configura la caché, habilita el chequeo automático de alineamiento, la escritura de páginas y el modo protegido. El registro de banderas, que por ejemplo, IF está controlada por cli, sti, iret. El TF genera una excepción de debug para permitir correr paso a paso. O el IOPL contiene el actual nivel de privilegio.



En el modo protegido de los procesadores Intel, utilizan el direccionamiento segmentado, en lugar de direccionamiento lineal. Mientras que el direccionamiento segmentado se refiere a que la memoria, tanto física como virtual, se dividen en bloque de 64K, ya que este es el valor máximo para el registro de puntero de instrucción IP. El registro IP trabaja con el registro de segmento de código (CS) para que apunte a la ubicación de memoria desde donde el microprocesador debe obtener la instrucción siguiente. La dirección IP utiliza 4 bytes para el direccionamiento de memoria.

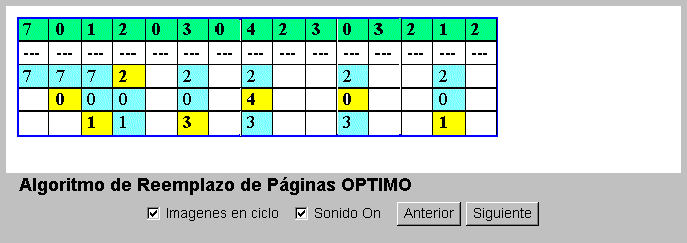
Sin embargo, dependiendo del tipo de procesador serán los modos de direccionamiento. Está el modo implícito que consiste que en un dato está implícito en la propia instrucción, por ejemplo, Set Carry, Set Interrupts o Set Direcction, encienden el flag correspondiente indicado en la misma instrucción. Convert Byte to Word (CBW) extiende el bit del signo del registro AL a AX. El direccionamiento inmediato, un dato a operar está inmediatamente después del opcode de la instrucción. El direccionamiento registro, consiste en que el dato está en un segundo registro. Mientras que el directo, se refiere a que la dirección del dato está en el campo de la dirección del opcode, por lo que se mueve el contenido de las direcciones al registro AX, pero en este caso sólo se mueven dos bytes. Y por último, el indirecto, en el cual un dato es especificado por medio de una combinación de registro índice y base, y puede hacerse un desplazamiento.

**Problema 2**

Linux es un sistema operativo, donde un procesador Intel, corre en modo protegido con el objetivo de proteger la memoria para que un proceso no tenga la oportunidad de tirar todo el sistema. Asimismo, que, no cargue los programas que se estén usando, ya que sólo carga una parte de ellos y, además, comparte la memoria entre los procesos para aumentar la velocidad y disminuir el uso de memoria, mientras que para la memoria caché, utiliza la memoria libre.

El esquema de gestión de memoria en el sistema operativo de Linux empieza por la memoria virtual, que se refiere a aplicar como un tipo de extensión de memoria RAM con el objetivo de que la memoria utilizable crezca de manera considerable. El direccionamiento de la memoria virtual utiliza una tabla de páginas que se divide en tres niveles. El primer nivel es el *directorio de página*, el cual se refiere a que cada uno de los procesos activos (los procesos activos deben encontrarse en la memoria principal), tienen un directorio de páginas único, y que cada entrada apunta a una página del directorio intermedio de páginas. El segundo nivel es el *directorio intermedio de página*, en donde se expanden múltiples páginas y cada entrada a este directorio apunta hacia a una página de la tabla de páginas. Por último, el tercer nivel es la *tabla de páginas*, la cual consiste en que cada entrada de la tabla apunta hacia una página de proceso virtual del proceso.

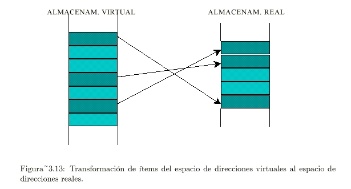
Para utilizar esta estructura de directorios, Linux lo ve como un conjunto de cuatro campos. El primer campo, que va desde la izquierda, se usa como el índice dentro del directorio de páginas. El siguiente campo, es el índice del directorio intermedio de páginas; el tercero hace referencia a la tabla de páginas y el último campo, indica el desplazamiento dentro de la página de la memoria.

Linux tiene un mecanismo donde su fin es asignar páginas y al mismo tiempo aumenta la eficiencia de cargar y descargarlas en la memoria principal. El *algoritmo de reemplazo* de páginas de Linux, que está basado en el algoritmo de Reloj, consiste en asociar un bit de uso y otros de modificación a cada una de las páginas de la memoria principal. Después, el bit de uso se remplaza por una variable de 8 bits y cada vez que se accede a la página, esta variable aumenta. Esta variable puede llamarse “edad”, ya que Linux la utiliza para hacer el cambio de página, es decir, Linux recorre periódicamente la lista completa de páginas de la memoria principal y les disminuye la variable edad a cada una de ellas. Entonces una página con edad “0” es considerada una página vieja, y eso quiere decir que tiene un buen tiempo que no ha sido referenciada por lo que se vuelve la indicada para ser remplazada. Sin embargo, si el valor de edad es muy alto, quiere decir que esa página ha sido referenciada muchas veces, por lo que no podría ser candidata para un remplazo.

El mecanismo de asignación de páginas usado para el manejo de la memoria virtual de usuario, son los fundamentos de asignación de memoria Kernel. Se maneja un sistema en base del algoritmo de colegas, donde se liberan y se asignan, en unidades, las páginas de la memoria del Kernel. Debido a que el tamaño mínimo de memoria que se puede reservar es de una página, la reserva de páginas no sería factible puesto que el núcleo necesita de pequeños fragmentos que se usarán por poco tiempo y que tienen tamaños distintos, es por esta razón que Linux usa un esquema llamado “asignación de láminas”.

El direccionamiento es una parte importante dentro del tema de manejo de memoria, pues influye tanto en el proceso como en el aprovechamiento del hardware y el rendimiento del sistema. Así que, el sistema operativo tiene una ayuda por parte del hardware para el manejo de la memoria, pues lo apoya con una *Memory Management Unit* que es la unidad encargada de realizar la segmentación y partición. Después, son bits reservados tanto en el selector de segmento y descriptores de página, y bits de protección. No obstante, el sistema es libre de usar o no estos recursos. Por otro lado, de la parte del procesamiento, es importante que se cumplan ciertas características como es la protección, la memoria de un proceso no se debe compartir con los demás, es decir, la memoria es diferente para cada uno de los procesos. Sin embargo, en Linux los procesos son separados en las direcciones de usuarios, ya que son propio del usuario, pero en un espacio de direcciones en Kernel, todos los procesos lo comparten. Otra característica, es la memoria virtual, la cual se refiere a que el proceso debe tener la ilusión de estar solo para facilitar el trabajo de los compiladores. Además, es necesario emplear técnicas de *swapping* y paginación por demanda, aprovechar la demanda memoria mediante técnicas de *copy-on-write*, un mapa sobre los ficheros de memoria. Y en general, mejorar el rendimiento del sistema aplicando diferentes métodos para asignar y liberar memoria.

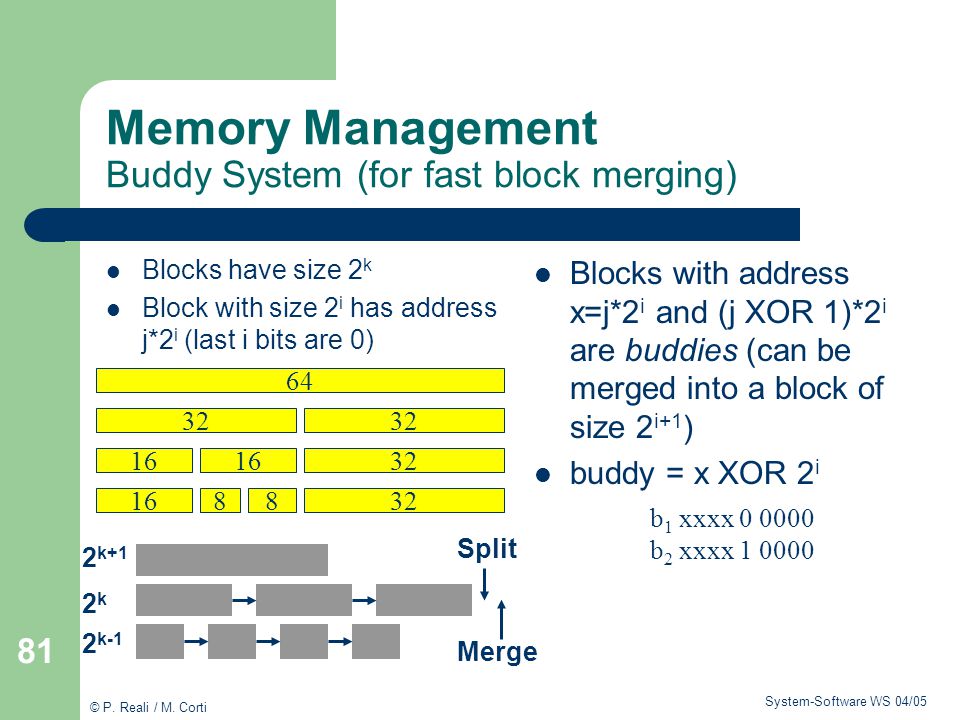
Existen tres tipos de direcciones: las *direcciones lógicas*, que son generadas por un proceso y cada una consiste en un selector de segmento y un desplazamiento que se refiera a la distancia entre el principio del segmento hasta la dirección actual; las *direcciones lineales,* generalmente representadas en hexadecimal, sirven para direccionar direcciones físicas de memoria; por último, las *direcciones físicas*, que hacen referencia a la memoria física.

La segmentación dentro de los sistemas Linux consiste en la traducción de dirección virtual a una dirección lineal. Cada segmento es un espacio contiguo de direcciones lineales con una dirección base, que es la dirección lineal de comienzo del mismo segmento; un límite, que especifica el tamaño en bytes o páginas, y finalmente sus atributos. Ahora, dependiendo del procesador será el registro de sus segmentos, y a su vez cada registro se enfoca a cosas distintas. Por ejemplo, el registro de segmento de código se refiere a un segmento que contiene las instrucciones del programa.

Sin embargo, la segmentación en Linux no es aprovechable por lo cual la usa limitadamente. En lugar de eso, se usa la paginación. La paginación es una técnica que traduce la dirección lineal a física, con esto revisa el tipo de acceso respecto a los derechos de acceso de la dirección lineal, pero si el acceso no es válido, se genera una excepción de falta de página. Esto puede ocurrir por las siguientes tres razones: la página no está presente, la página está sin asignar o el acceso es incorrecto.

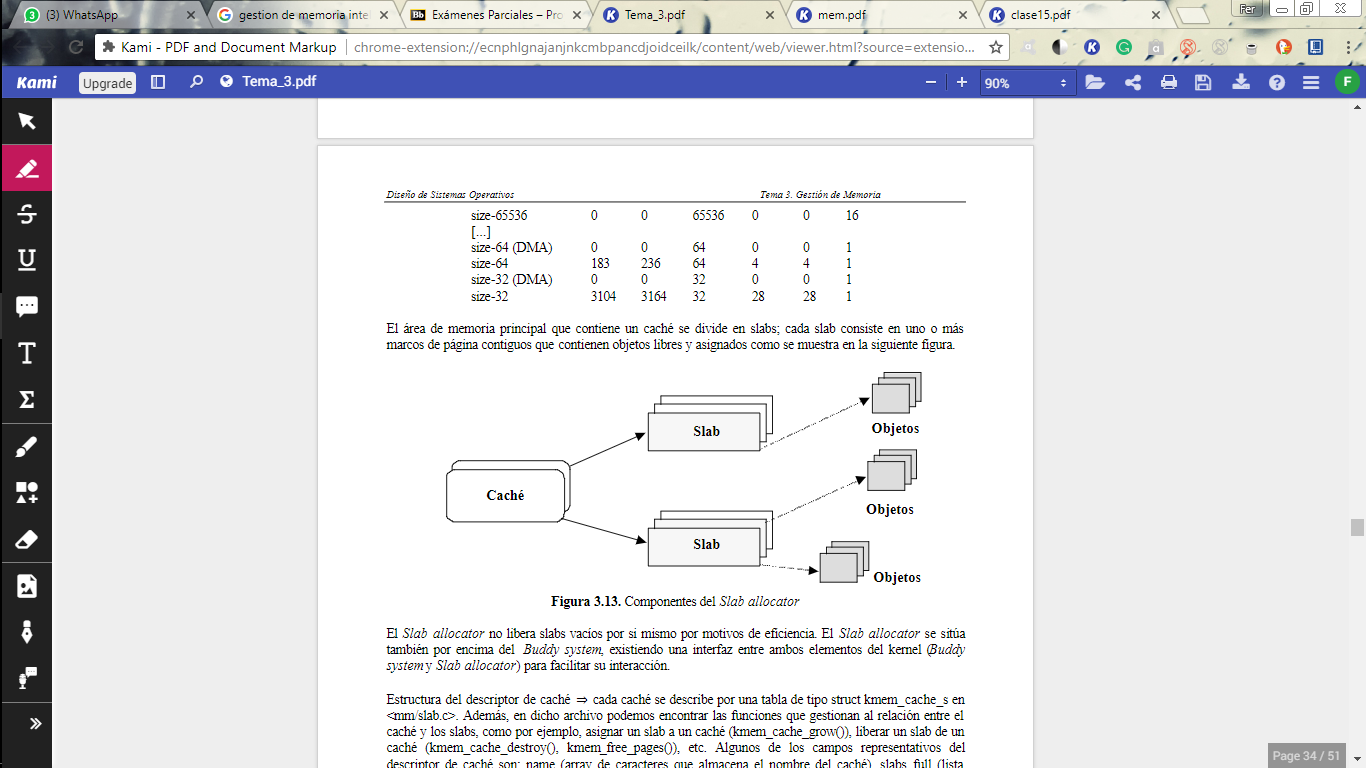
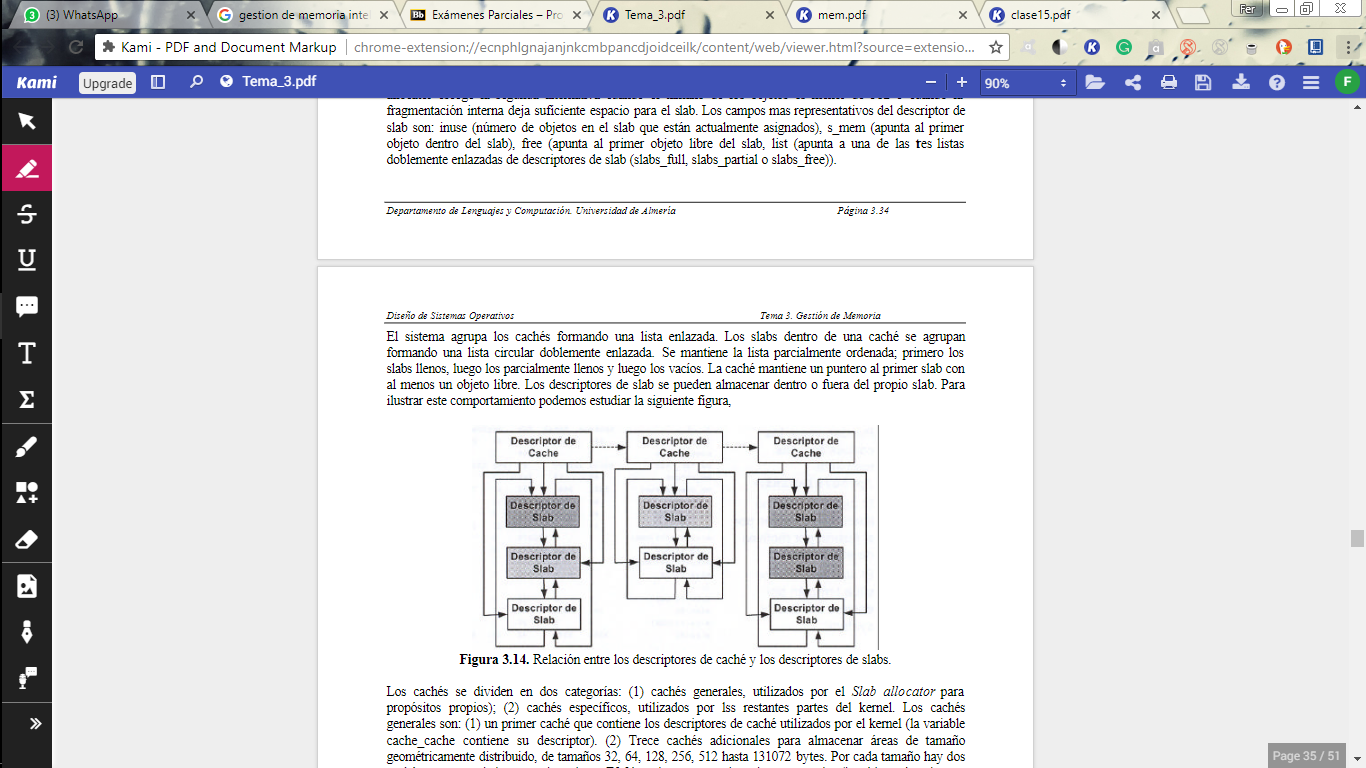
Los estados de la página pueden ser: disponible, que se refiere a las páginas que actualmente no se están usando en el proceso; reservada, que es conjunto de páginas que están juntas pero que el gestor memoria virtual separa para un proceso y asignada, que son las páginas que el gestor de memoria virtual ha reservado espacio en el fichero de paginación. La diferencia entre el estado de reservada y asignación es muy útil ya que reduce la cantidad de espacio del disco que debe guardarse para un proceso en específico, manteniendo espacio libre para otros procesos. Además de que permite que un proceso pida una cantidad de memoria y se le pueda dar de manera rápida.

El kernel puede solicitar memoria de tres formas: directamente al *Buddy system*¸ para asignaciones genéricas de grupos de marcos de página potencia de dos. El sistema de colegas o *buddy system*, se encarga de asignar grupos de marcos páginas contiguos, por lo que su objetivo es evitar la fragmentación externa. Para evitar esto, existen dos posibilidades: utilizar la unidad de paginación para agrupar marcos de página de dispersos en direcciones lineales contigua o desarrollar un sistema que controle los marcos de página contiguos y evite en lo posible dividir un bloque libre grande para una asignación pequeña. Sin embargo, el kernel opta por la segunda opción porque en ocasiones es necesario que los marcos de página sean contiguos, ya que direcciones lineales contiguas no son suficientes para satisfacer la petición. Además, no se modifican las tablas de páginas del kernel.

El kernel mantiene una lista de grupos de marcos de página, siendo éstos de tamaño fijo y se refieren a marcos de página contiguas en memoria. Buddy system se refiere a que cada petición de asignación, se usa la lista que no está vacía que contiene los grupos de marcos de página de tamaño inmediatamente superior al tamaño especificado y se selecciona un grupo de páginas de esta lista. Este grupo se divide en dos partes: los marcos de página correspondientes al tamaño de memoria especificado, y el resto de marcos de página que siguen disponibles y pueden insertarse en otras listas. Al liberar un grupo de marcos de página, el kernel intenta fusionar este grupo con los grupos disponibles con el objetivo de obtener un grupo disponible de tamaño máximo. Para simplificar la asignación y liberación de un grupo de marcos de página, el kernel sólo permite asignar grupos de páginas cuyo tamaño sea predeterminado y corresponda a los tamaños gestionados en las listas.

La otra forma de solicitar memoria es por medio del *Slab allocator*, que se usa para objetos frecuentemente usados. El sistema *Slab Allocator* funciona para asignaciones pequeñas de memoria, ya que el *Buddy System* se encarga de asignar grandes cantidades desperdiciando memoria, fragmentación interna, por lo que se definen 13 listas de áreas de memoria donde guarda objetos de tamaños geométricamente distribuidos depende más de ser potencia de dos que del tamaño de los datos a almacenar. Entonces el *Slab allocator* busca acelerar las asignaciones y liberaciones de objetos y evitar fragmentación interna. Se basa en que el kernel pide y libera áreas de memoria pequeñas, mantienen una serie de cachés que gestionan slabs de objetos del mismo tipo, además, un slab es una serie de marcos de página contiguos en lo que se almacenan los objetos del mismo tipo. Por otro lado, cuando se pide memoria para un objeto se obtiene de la caché correspondiente, ahorrando tiempo y evitando la fragmentación interna. Después, cuando se libera un objeto, no se desasigna realmente la memoria, sino que se marca la entrada en el slab como disponible, pero si no hay espacio en ningún slab de la caché par a un nuevo objeto, se crea un nuevo slab. Sin embargo, no libera slabs vacíos por sí mismo por motivos de eficiencia.

El área de memoria principal que contiene un caché se divide en slabs, cada slab consiste en uno o más marcos de página contiguos que contienen objetos libres y asignados.

**Bibliografía**

**Intel® 64 and IA-32 Architectures Software Developer Manuals | Intel® Software**

*Intel® 64 and IA-32 Architectures Software Developer Manuals | Intel® Software*. (2017). *Software.intel.com*. Retrieved 24 October 2017, from <https://software.intel.com/en-us/articles/intel-sdm>

### Capítulo 7. Administración de Memoria

#### *Capítulo 7. Administración de Memoria*. (2017). *Tldp.org*. Retrieved 24 October 2017, from <http://www.tldp.org/pub/Linux/docs/ldp-archived/system-admin-guide/translations/es/html/ch07.html>

### Urbina, C.

#### Urbina, C. (2011). *Gestión de memoria Linux Windows*. *Os-its.blogspot.mx*. Retrieved 24 October 2017, from <http://os-its.blogspot.mx/2011/10/gestion-de-memoria-linux-ubuntu.html>

### Modos de direccionamiento utilizado por los Procesadores Intel®

#### *Modos de direccionamiento utilizado por los Procesadores Intel®*. (2017). *Intel*. Retrieved 25 October 2017, from <https://www.intel.es/content/www/es/es/support/articles/000007194/processors.html>

### Gestión de memoria

#### *Diseño de Sistema Operativos Tema3. Gestián de memoria* (2017). *Intel*. Retrieved 25 October 2017, from <https://w3.ual.es/~acorral/DSO/Tema_3.pdf>

### Procesador IA-32 - Gestión de Memoria

#### *Procesador IA-32*. (2017). *Intel*. Retrieved 25 October 2017, from <https://www.dc.uba.ar/materias/oc2/2017/c1/Teoricas/mem.pdf>

### Intel 8086 y 8088

#### *Intel 8086 y 8088*. (2017). *Es.wikipedia.org*. Retrieved 25 October 2017, from https://es.wikipedia.org/wiki/Intel\_8086\_y\_ 8088#Modos\_de\_direccionamiento