



INDICE	PÁGINAS
⇒ INTRODUCCIÓN	119
⇒ CLASIFICACIÓN	119
⇒ TIPOS DE DIRECCIONES	120
⇒ MEMORIA REAL	121
⇒ ORGANIZACIÓN DE LA MEMORIA DE UNA MICROCOMPUTADORA	121
⇒ PROTECCIÓN	122
⇒ REUBICACIÓN	123
⇒ TÉCNICAS DE INTERCAMBIO	
⇒ SWAPPING	124
⇒ OVERLAYING	125
⇒ ASIGNACIÓN DE PARTICIONES MÚLTIPLES	127
⇒ REGISTRO LÍMITE	128
⇒ REGISTRO BASE Y LÍMITE	128
⇒ MULTIPROGRAMACIÓN	129
⇒ MULTIPROGRAMACIÓN CON PARTICIONES FIJAS	130
⇒ MULTIPROGRAMACIÓN CON PARTICIONES VARIABLES	131
⇒ COMPACTACIÓN	132
⇒ MEMORIA VIRTUAL	
⇒ ORGANIZACIÓN DEL ALMACENAMIENTO	134
⇒ MODO DE FUNCIONAMIENTO (PAGINACIÓN-SEGMENTACIÓN)	136
⇒ VENTAJAS Y DESVENTAJAS	140
⇒ RENDIMIENTO	141



GESTION DE MEMORIA

La organización y administración de la “memoria principal”, “memoria primaria” o “memoria real” de un sistema ha sido y es uno de los factores más importantes en el diseño de los sistemas operativos.

Harvey M. Deitel

INTRODUCCIÓN

La memoria tiene un carácter central en el funcionamiento de un sistema de computación moderno. Tanto la CPU como los sistemas de E/S intercalan con la memoria. La memoria es una amplia tabla de palabras o bytes, cada una de las cuales tienen su propia dirección. La interactividad se obtiene mediante una secuencia de lecturas y escrituras en direcciones específicas de memoria. La CPU extrae y almacena información de la memoria.

CLASIFICACIÓN

Esta clasificación de la memoria nos permitirá identificar a los distintos espacios de memoria que debe administrar un sistema operativo a partir de su ubicación geográfica dentro de un sistema de computación como en el modo de direccionamiento que utilizará para acceder a los diferentes espacios.

Las memorias que utiliza un sistema operativo para la administración de los procesos, como ya sabemos son lugares de almacenamiento, estos lugares, desde el punto de vista físico pueden ser electrónicos o magnéticos, desde el punto de vista de modo uso se las cataloga como la memoria principal que es del tipo electrónico y la memoria secundaria o auxiliar que es del tipo magnético. A la memoria principal se la considera como la memoria real y a la secundaria como memoria virtual.

Clasificación:

- **POR EL MODO DE DIRECCIONAMIENTO**
 - Indirecto: MEMORIA VIRTUAL
 - Tipos
 - Extensión de la memoria RAM.
 - Memoria secundaria o auxiliar.
 - Directo: MEMORIA REAL



- **POR EL MEDIO DE ALMACENAMIENTO**
 - **Electrónico:**
 - MEMORIA PRINCIPAL
 - RAM
 - ROM
 - MEMORIA INTERMEDIA
 - CACHE
 - **Magnética:**
 - MEMORIA SECUNDARIA O AUXILIAR.

TIPOS DE DIRECCIONES

Existen 2 tipos de direcciones: las direcciones lógicas y las direcciones físicas, el usuario genera únicamente direcciones lógicas, el sistema operativo sabe algo más y puede acceder directamente en modo monitor. Toda la información pasada desde el programa de usuario al sistema operativo tiene que reubicarse explícitamente por parte del software del sistema operativo antes de utilizarla.

Esta necesidad es particularmente cierta en el caso de las direcciones dadas a los dispositivos de E/S, el usuario suministra las direcciones lógicas, el sistema operativo tiene que convertirlas en direcciones físicas antes de ser utilizadas.

En la mayoría de los casos un programa pasa por varias etapas antes de ser ejecutado, durante esas etapas las direcciones pueden representarse de varias maneras:

- Cuando se escribe el programa fuente las direcciones son generalmente simbólicas, como por ejemplo una instrucción o la asignación de un valor a una variable, ocupan simbólicamente un espacio en memoria.
- Cuando se compila el programa liga estas direcciones simbólicas a direcciones reubicables, o sea, que la compilación generó el objeto, que ya determinado el tamaño que le deberá asignar el sistema operativo a partir de la modalidad de ejecución del programa compilado.
- El montador de enlaces (linkeador) ligará a su vez estas direcciones reubicables a direcciones absolutas. Cada ligadura es un acoplamiento de un espacio de direcciones a otro. Finalmente, un programa tiene que acoplarse a direcciones absolutas y cargarse para ejecutarse. A medida que se ejecuta, accede a las instrucciones y datos de la memoria generando estas direcciones absolutas. Finalmente el programa termina, se declara disponible su espacio en la memoria y puede ejecutarse el programa siguiente.

Las direcciones de memoria a partir de la ejecución de un programa se van reubicando dinámicamente



MEMORIA REAL

Se denomina memoria real al espacio de direcciones físicas perteneciente a la RAM, en donde los programas de usuario estarán almacenados durante el tiempo que se encuentren activos, o sea, realizando un proceso.

ORGANIZACIÓN DE LA MEMORIA DE UNA MICROCOMPUTADORA

Aunque toda la memoria es igual, según su lugar en el mapa de memoria adquiere diversos nombres. Cada programa necesita uno o varios tipos de memoria. Por esta razón debemos aprender a configurar el ordenador de acuerdo a los requerimientos de cada aplicación.

- **Memoria Base o Convencional:**

Comprende la porción entre 0K y 640K. Esta memoria es aprovechada sin problemas por la totalidad de órdenes del sistema y las aplicaciones. Debido a ciertas limitaciones del microprocesador y del propio sistema operativo, otras zonas de memoria tienen un acceso mucho más difícil. Por tanto, todos los programas que admitan otro tipo de memoria, deben trasladarse para conseguir la mayor cantidad posible de memoria convencional. La zona entre 0K y 64K se denomina también memoria baja.

- **Memoria Superior (Upper Memory Area - UMA)**

Es la zona situada entre los 640K y los 1024K (1Mb). Esta memoria es aprovechada principalmente por el sistema de computación. Como la memoria ROM es lenta, se suelen copiar a esta zona, las rutinas de la ROM-BIOS y así conseguir una velocidad global mucho mayor. También se utiliza para almacenar los caracteres impresos en la pantalla en cada momento, los controladores de dispositivos (drivers) y programas del tipo residente (TSR).

- **Memoria Alta (High Memory Area - HMA)**

Es la memoria comprendida entre 1024K y 1088K. Son los primeros 64 K de la extensión de memoria. En un equipo que cuente con este espacio de memoria, parte del sistema operativo puede estar almacenado en esta parte de la memoria, esto dejará más memoria convencional disponible para los programas.



- **Memoria Expandida (Expanded Memory Specification - EMS)**

Es una técnica software utilizada para acceder a la memoria por encima de 1 Mb. Para emplear memoria expandida se precisa un administrador de memoria expandida. Como las aplicaciones sólo son capaces de acceder a la memoria inferior de 1Mb, el administrador se encarga de copiar, según se necesiten, los datos por encima de 1Mb a un marco de página de 64K situado en memoria superior.

Como solamente se puede trabajar con 64K de información a la vez, es necesario copiar continuamente datos desde la memoria expandida (más de 1Mb) a la memoria superior y viceversa. Otra solución más rápida y eficiente es la llamada memoria extendida.

- **Memoria Extendida (Extended Memory Specification - XMS)**

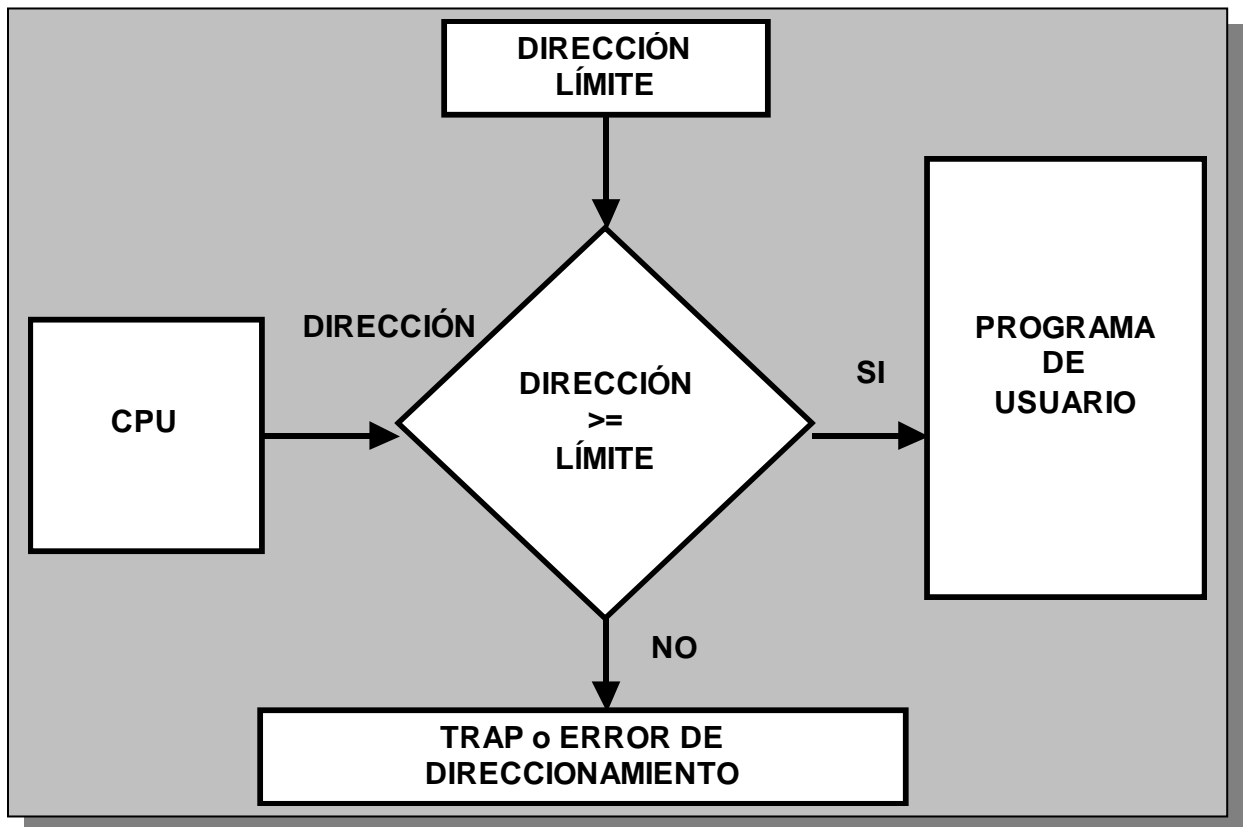
Es la memoria por encima de 1Mb. Para poder alcanzar esta región, el microprocesador debe trabajar en un modo llamado modo protegido. Aunque en el DOS, no es capaz de operar en este modo, la mayoría de las aplicaciones sobre la base de este sistema operativo emplean diversas técnicas para acceder a memoria extendida.

División de Memoria Principal

EXTENSIÓN DE MEMORIA	1088 a ... Kb.
MEMORIA ALTA	1024 a 1088 Kb.
MEMORIA SUPERIOR	640 a 1024 Kb.
MEMORIA CONVENCIONAL	0 a 640 Kb.

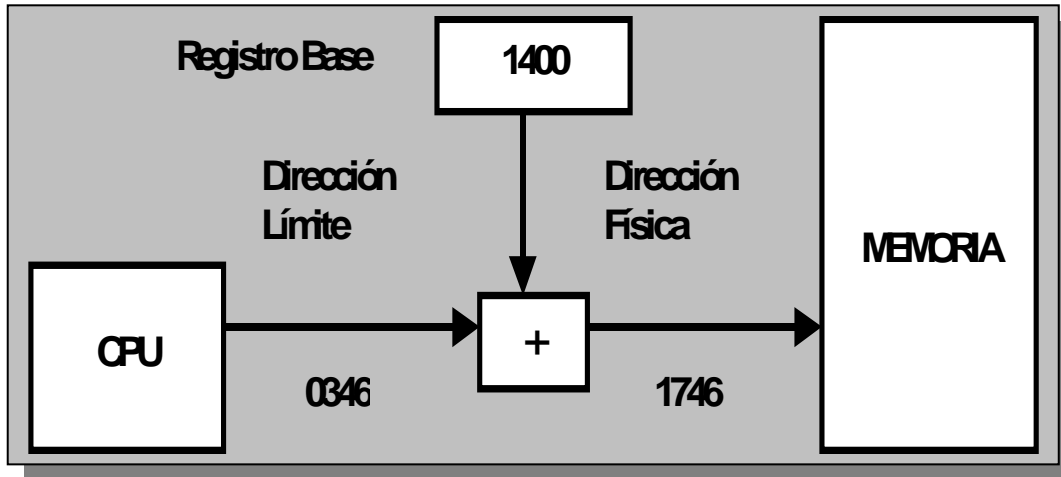
PROTECCIÓN

Un programa de un usuario se ejecuta en la parte alta de la memoria, por ese motivo debemos proteger al monitor de los cambios accidentales o intencionales que puedan ocasionar los programas. Esta protección tiene que ser suministrada por el hardware y puede implementarse de varias maneras. Cada dirección (instrucción o dato) generada por un programa es comparada con una dirección límite o separadora, si esta dirección generada es mayor o igual al límite, se trata de una dirección correcta, y es enviada a la memoria. Si la dirección generada es menor que el límite, se intercepta y se genera un trap o error de direccionamiento, en ese caso el sistema operativo terminará la ejecución del programa con un mensaje de error, dejando libre la memoria.



REUBICACIÓN

La reubicación dinámica de los programas depende exclusivamente del tipo de registro utilice el sistema operativo para la asignación de los distintos espacios de memoria. Un problema a considerar en la carga de los programas a la memoria, y teniendo presente que el espacio de direccionamiento en una computadora empieza en la dirección 0, y que la primer dirección generada por un programa de usuario nunca es 0, sino que es la primer dirección tras el límite. La dirección límite se denomina registro límite, y el registro límite es el registro base o registro de reubicación. El valor del registro se añade a la dirección generada por el proceso en el momento que la envía a la memoria, por ejemplo, si el límite se encuentra en la dirección 1400, y el programa envía la primer dirección pensando que es 0, cómo la dirección límite es 1400, se reposiciona dinámicamente en esa dirección. Si después el programa quiere acceder a la dirección 346, suma el valor del límite más el valor de la dirección generada y se reubica en la dirección 1746.



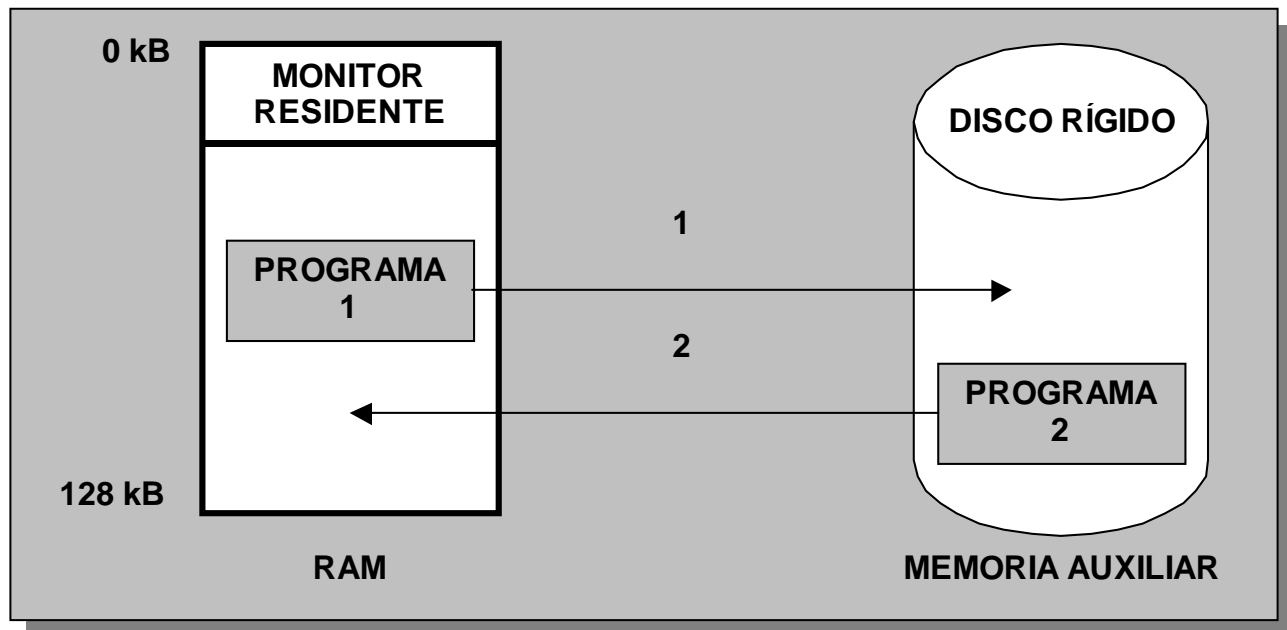
Un usuario nunca sabe las direcciones físicas reales. El usuario genera solamente direcciones lógicas. El hardware de planificación de memoria convierte las direcciones lógicas en direcciones físicas.

Toda la información pasada desde un programa de usuario al sistema operativo tiene que ser reubicada por el software del sistema operativo antes de poder ser utilizada.

TÉCNICAS DE INTERCAMBIO

SWAPPING¹: El swapping requiere de memoria auxiliar. La memoria auxiliar suele residir en disco. Tiene que ser lo suficiente grande como para almacenar las copias de todas las imágenes de memoria de todos los usuarios, y tiene que ofrecer acceso directo a todas estas imágenes de memoria. El swapping es la técnica por la cual el sistema operativo permite que dos o más programas se alternen en el uso de la misma área de memoria, mientras un programa ejecuta en memoria principal, los restantes son almacenados en estado de suspensión en un medio auxiliar de almacenamiento, en algún momento el programa de la memoria principal se suspende, se almacena en un medio auxiliar y otro programa es cargado en la memoria principal para su ejecución.

¹ Peterson, J. L., Silberschatz, A., *Sistemas Operativos: Conceptos fundamentales*, Editorial Reverté S. A., Barcelona, España, 1989, pág. 162.



1. Reincorporación a la memoria auxiliar (programa inactivo).
2. Descargar de la memoria auxiliar (programa activo).

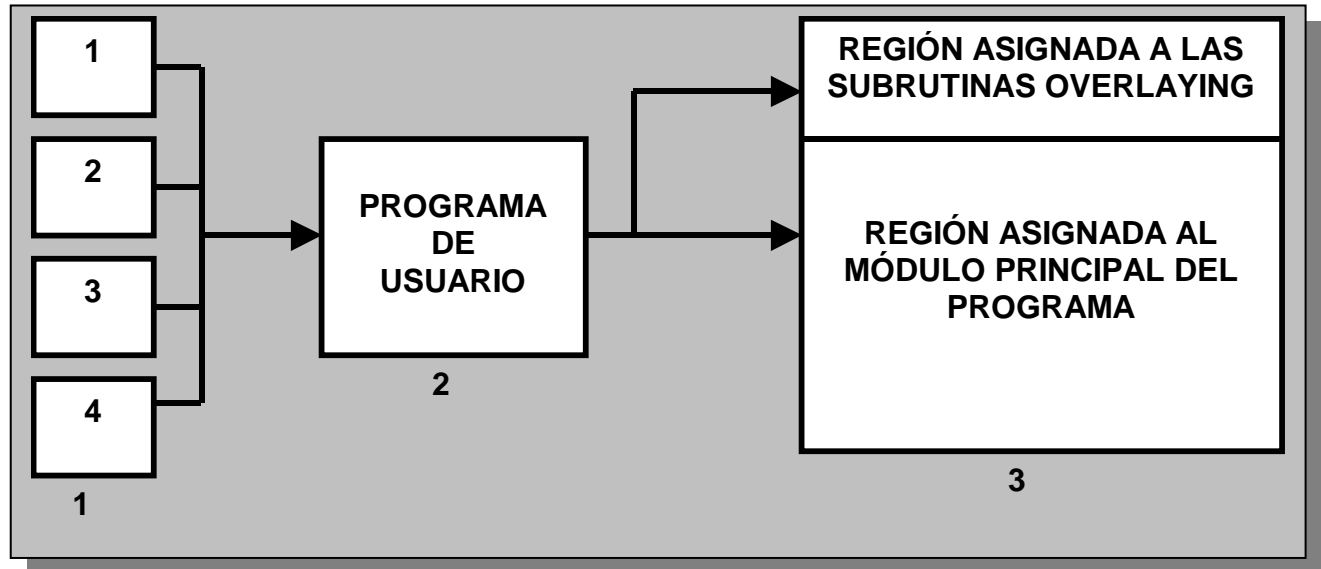
Si el proceso de swapping ocurre a intervalos frecuentes y regulares la computadora puede aparecer como si estuviera ejecutando muchos programas al mismo tiempo y permitiendo de este modo que el programa use el sistema completo, de esta forma el swapping es invisible al programador y es manejado por el sistema operativo a través del componente de administrador de memoria (storage allocator).

OVERLAYING²: El overlaying es la técnica por la cual ciertas partes de un programa de usuario único, puede cambiar en memoria. Esta técnica es útil para situaciones en las cuales el programa es físicamente más largo que la memoria disponible.

Cuando el programa ha sido escrito con subrutinas de overlaying permite al programador llevar una subrutina dada a la memoria sólo cuando ésta se necesita.

² <http://www.tau.org.ar/base/lara.pue.udlap.mx/sistoper/capitulo4.html>
www.uco.es/~el1orlom/docs/STRtema4.pdf

"Real-Time Computer Control". Stuart Bennett. Prentice Hall. 1993.



1. Subrutinas overlaying (módulos del programa).
2. Módulo principal del programa.
3. Área de memoria asignada al programa.

A diferencia del swapping, el usuario puede controlar el funcionamiento del programa en donde se especifican que rutinas pertenecen al módulo principal del programa, y cuales pertenecen a cada una de las regiones de overlaying.

Una vez que el programa está en ejecución, las operaciones de overlaying son realizadas automáticamente por el storage allocator.

DIFERENCIAS ENTRE SWAPPING Y OVERLAYING

SWAPPING	OVERLAYING
<ul style="list-style-type: none">• Iniciada por el SISTEMA OPERATIVO• Bidireccional (un programa va hacia el disco y otro viene desde el disco).• El programa completo y sus datos realizan el swapped.• El programa activo es aquel que se encuentra en la memoria RAM.	<ul style="list-style-type: none">• Iniciada por el programa.• Unidireccional (todos los módulos overlaying se mantienen en disco y se cargan en memoria cuando se necesitan, borrando el contenido anterior de la memoria reservada para el intercambio).• Sólo las subrutinas son overlayed, nunca los datos.



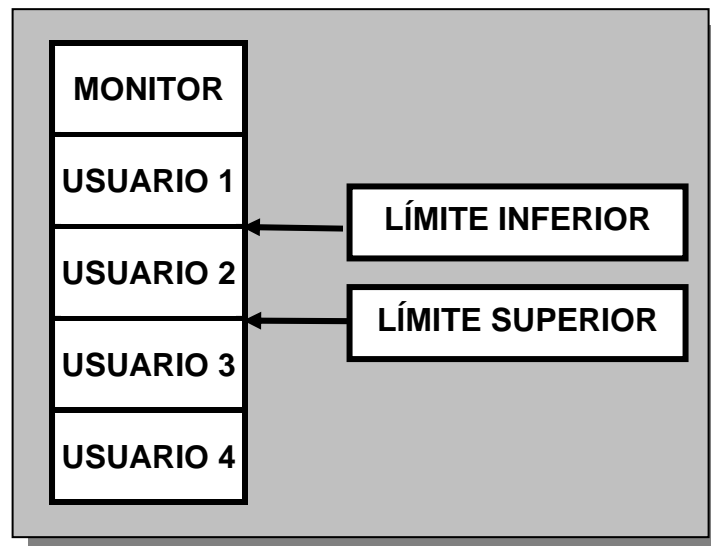
ASIGNACION DE PARTICIONES

La configuración de la memoria que resulta del intercambio es efectivamente la misma que la configuración de la memoria para la multiprogramación, hay más de un programa en la memoria al mismo tiempo. En el caso de la multiprogramación, la CPU pasa rápidamente de un programa a otro. El problema para la gestión de memoria es la asignación de memoria para los programas que se estén por ejecutar.

Por ese motivo la memoria se divide en particiones o regiones. Cada partición puede contener un programa a ejecutar. Así el grado de multiprogramación está limitado por el número de particiones.

Son posibles dos esquemas principales para la gestión de memoria, cada uno divide a la memoria en un cierto número de particiones o regiones. Estos esquemas son la asignación contigua de particiones fijas y la asignación contigua de particiones variables.

Sin embargo se necesita de la posibilidad de proteger el espacio que se encuentra antes o después de la ejecución de un programa, esto se puede obtener utilizando dos tipos de registros que constituyen los límites superior e inferior de las direcciones que generen legalmente los programas de los usuarios, estos registros pueden definirse de dos maneras, como registro límite o registro base y límite:



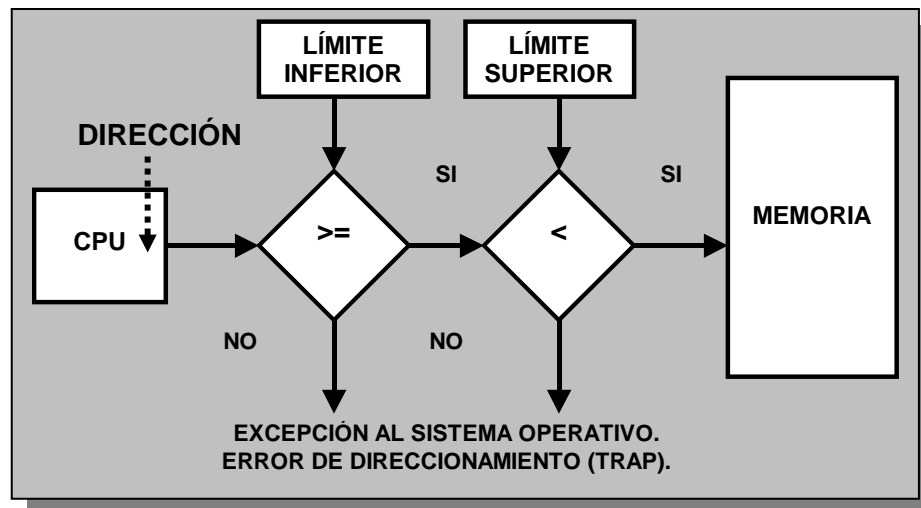


- **Registros Límite:**

Son los valores de la menor y la mayor dirección física.

Las direcciones legales del usuario van del límite inferior al límite superior, por ejemplo, límite inferior = 100.040 y límite superior = 174.640.

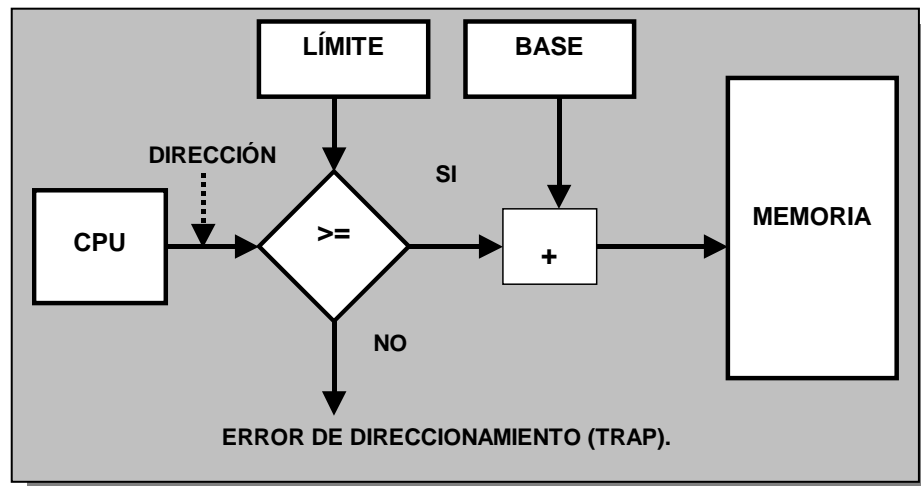
Los registros límite exigen la reubicación estática, en el momento de la carga. Cada dirección lógica tiene que ser mayor o igual que el límite inferior y menor que el límite superior.



- **Registros base y límite:**

Son los valores de la dirección física más baja y del margen de direcciones lógicas, por ejemplo, base = 100.040 y límite = 74.600.

Con los registros base y límite, cada dirección lógica tiene que ser inferior y se reubica dinámicamente, sumándole el valor al registro base. Esta nueva dirección se envía a la memoria.





MULTIPROGRAMACIÓN

Al introducirse la multiprogramación, el problema de la utilización de memoria se agrava, dado que al tener que alojar a varios programas en memoria, hacia necesario evitar los desperdicios de memoria (ya que los requerimientos de memoria se agrandan considerablemente) contemplando que los distintos programas requerían distintas cantidades de memoria.

Otro problema planteado por la multiprogramación, es que cada programa podía ubicarse en distintas posiciones de memoria cada vez que fuera ejecutado, esto fue resuelto simplemente con el concepto de reubicabilidad de los programas, expresando las direcciones en términos de una dirección absoluta dentro del programa, comenzando desde cero (desplazamiento) y un índice que, cargado en el momento de ejecución, permite incorporar la dirección de comienzo de programa en la memoria (registro base). De esta manera las direcciones son relativas al comienzo del programa, cualquiera sea su ubicación en la memoria en el momento de ejecución.

Este problema fue solucionado a través de los compaginadores y compiladores, ya que al generar las direcciones en lenguaje objeto, éstas son expresadas en términos de base y desplazamiento (límite).

Este esquema se basa en la división de la memoria en particiones, cada una destinada a alojar a un programa distinto. Debido a esto la multiprogramación plante los siguientes inconvenientes:

1. Se necesita mayor cantidad de memoria, dado que por un lado el sistema operativo es más complejo y ocupa más cantidad de memoria y por otro se necesita el espacio adecuado para alojar en memoria una determinada cantidad de programas que permitan explotar convenientemente las ventajas de la multiprogramación.
2. Se necesitan procedimientos especiales de protección de memoria para evitar que accidental o intencionalmente un programa interfiera con otro (utilice o destruya la parte de memoria asignada a otro programa) o inclusive con el sistema operativo.
3. Otro punto a tener en cuenta se relaciona con la **Fragmentación de Memoria** que puede existir, a partir del aprovechamiento parcial o total del espacio de memoria que asignada a un programa, en este caso pueden ocurrir dos cosas, que el espacio que el sistema operativo le asigna a un programa no sea ocupado en su totalidad, generándose de esta forma la **fragmentación interna** de memoria o que el sistema operativo quiera cargar un programa en memoria y no tenga espacio suficiente, generándose en este caso una **fragmentación externa**.



La asignación de particiones reales, nos brinda 2 grandes variantes:

1. Multiprogramación con particiones de tamaño fijo.
2. Multiprogramación con particiones de tamaño variable.

MULTIPROGRAMACIÓN CON PARTICIONES FIJAS

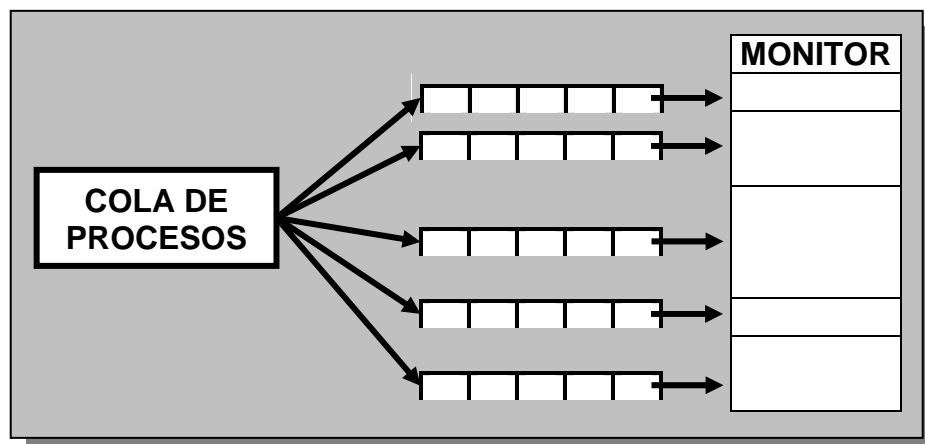
Los sistemas de un solo usuario desperdician gran cantidad de recursos computacionales debido a que:

- Cuando ocurre una petición de E/S la CPU normalmente no puede continuar el proceso hasta que concluya la operación de E/S requerida.
- Los periféricos de E/S frenan la ejecución de los procesos ya que comparativamente la CPU es varios órdenes de magnitud más rápida que los dispositivos de E/S.

Las “particiones” del almacenamiento principal:

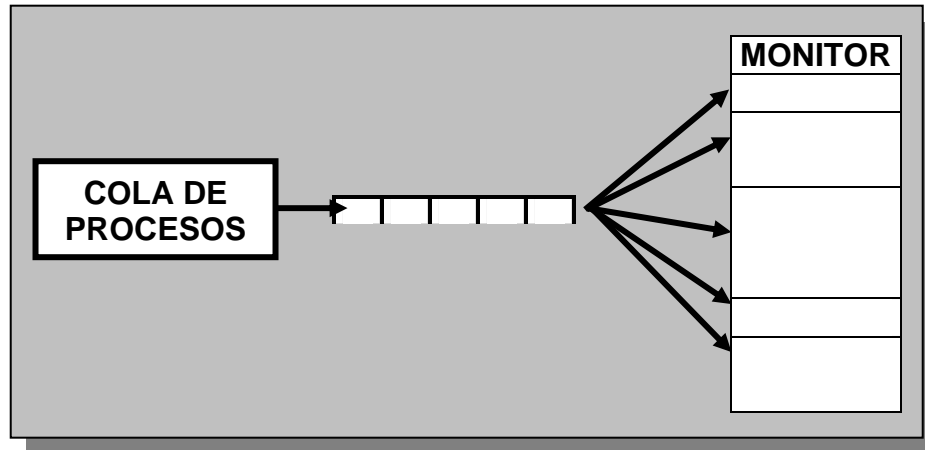
- Son de tamaño fijo.
- Alojan un proceso cada una.
- La CPU se cambia rápidamente entre los procesos creando la ilusión de simultaneidad.

Se puede presentar dos esquemas diferentes para la multiprogramación con particiones de tamaño fijo, el primero de ellos plantea la posibilidad de que exista una cola de procesos para cada partición.



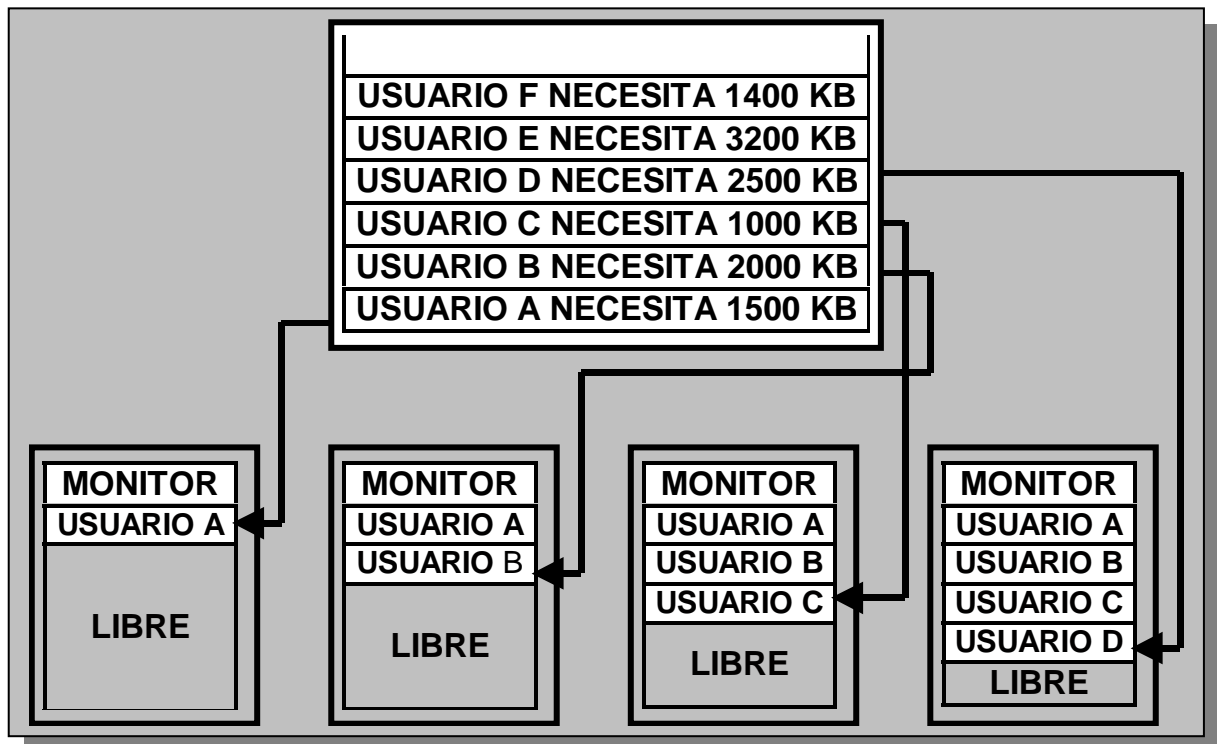


El segundo esquema brinda la posibilidad de que los procesos que estén preparados para ejecutarse formen una única cola de procesos que pueda utilizar cualquiera de las particiones.



MULTIPROGRAMACIÓN CON PARTICIONES VARIABLES

Los procesos ocupan tanto espacio como necesitan, pero obviamente no deben superar el espacio disponible de memoria





No hay límites fijos de memoria, es decir que la partición de un trabajo es su propio tamaño.

Se consideran “esquemas de asignación contigua”, dado que un programa debe ocupar posiciones adyacentes de almacenamiento.

Los procesos que terminan dejan disponibles espacios libres de memoria principal llamados “agujeros”:

- Pueden ser usados por otros trabajos que cuando finalicen dejen espacios libres de memoria de menor tamaño.
- En sucesivos pasos los espacios libres son cada vez más numerosos pero más pequeños, por lo que se genera un desperdicio de memoria principal.

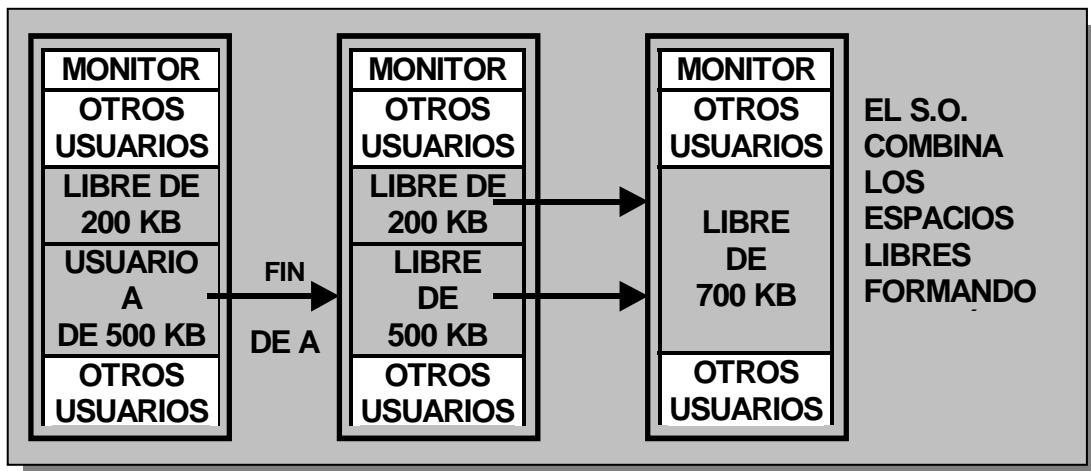
COMPACTACIÓN

Consiste en fusionar los espacios libres adyacentes para formar un espacio libre de direcciones continuas más grande.

Se puede hacer cuando un trabajo termina y el almacenamiento que libera tiene límites con otros espacios libres.

Puede ocurrir que los espacios libres separados distribuidos por todo el almacenamiento principal constituyan una cantidad importante de memoria:

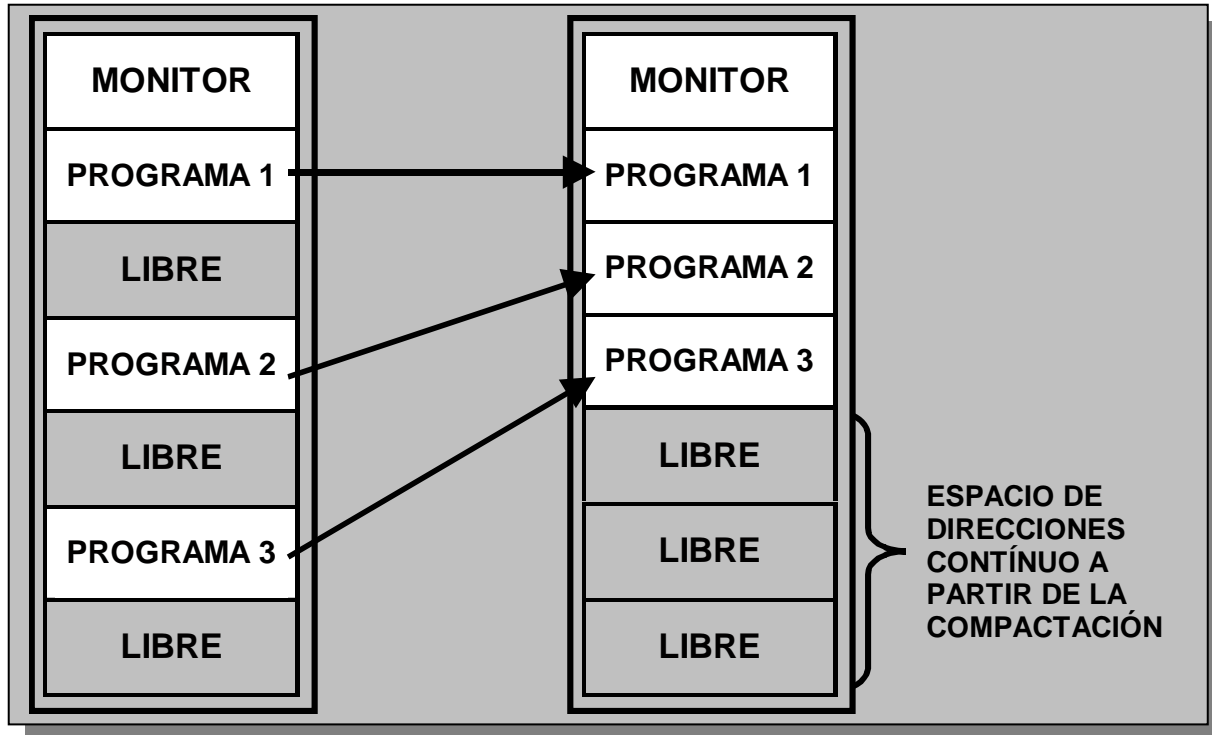
- Podría ser suficiente (el total global disponible) para alojar a procesos preparados para ejecutarse que se encuentren en espera de memoria principal disponible.
- Podría no ser suficiente ningún área libre individual.





Principales desventajas de la compactación:

- Consume recursos del sistema.



- El sistema debe detener todo mientras efectúa la compactación, lo que puede afectar los tiempos de respuesta.
- Implica la reubicación de los procesos que se encuentran en la memoria:
 - La información de reubicación debe ser de accesibilidad inmediata.
- Una alta carga de trabajo significa mayor frecuencia de compactación que incrementa el uso de recursos.

Como resumen, podemos decir que existen módulos de memoria asignada en forma fija a determinada partición (puede ser una cantidad variable por partición) y otros de libre asignación a la partición que los necesite y que luego de terminado el programa, el sistema operativo podría asignarlo a la misma partición o a otra.

La utilización de particiones de tamaño variable admite, a su vez, dos posibilidades: cantidad fija o variable de particiones.



En ambos casos el tamaño de cada partición es el necesario para alojar el programa que se está ejecutando en ella, pero en el caso de la cantidad fija de particiones, la cantidad de programas que integran la mezcla de multiprogramación tiene como máximo el número de particiones existentes; en cambio, cuando la cantidad de particiones es variable, la cantidad de programas alojadas en memoria en ejecución dependerá de la capacidad de almacenamiento de la memoria y de la cantidad requerida por cada programa.

Estas variantes, como se podrá apreciar, solucionan sólo en parte los problemas de desaprovechamiento de memoria del método de particiones de tamaño fijo, dado que todavía se requiere que el programa está en memoria en posiciones consecutivas.

MEMORIA VIRTUAL

ORGANIZACIÓN DEL ALMACENAMIENTO

“Almacenamiento virtual” significa la capacidad de direccionar un espacio de almacenamiento mucho mayor que el disponible en el almacenamiento primario de determinado sistema de computación.

Esta tecnología apareció en 1960 en la Universidad de Manchester (Inglaterra), en el sistema “Atlas”.

Los métodos más comunes de implementación son mediante:

- Técnicas de “paginación”.
- Técnicas de “segmentación”.
- Una combinación de ambas técnicas.

Las direcciones generadas por los programas en su ejecución no son, necesariamente, aquellas contenidas en el almacenamiento primario (memoria real), ya que las direcciones virtuales suelen seleccionarse dentro de un número mucho mayor de direcciones que las disponibles dentro del almacenamiento primario.



La evolución en las organizaciones de almacenamiento puede resumirse como sigue:

- **Real:**
 - Sistemas dedicados a un solo usuario.
- **Real:**
 - Sistemas de multiprogramación en memoria real:
 - Multiprogramación en partición fija:
 - Absoluta.
 - Relocalizable (reubicable).
 - Multiprogramación en partición variable.
- **Virtual:**
 - Multiprogramación en almacenamiento virtual:
 - Paginación pura.
 - Segmentación pura.
 - Combinación paginación / segmentación.

La memoria virtual es un espacio de direcciones cuyo tamaño máximo está determinado por el esquema de direccionamiento que utilice la computadora, es decir, por la cantidad máxima de bytes posibles a direccionar que depende de la cantidad de bits o dígitos binarios utilizados para expresar una dirección.

Si cada byte tiene una dirección distinta, la cantidad de bytes que pueden ser referenciados dependerá de la cantidad de bits que se utilicen para expresar esa dirección.

Por ejemplo: con 1 bit sólo pueden ser referenciados 2 bytes, el 0 y el 1. Con 8 bits se podrán direccionar 256 bytes, con 16 serán 65.536 y con 24 serán 16.777.216 bytes.

El espacio de almacenamiento que puede ser accedido directamente por una computadora se denomina memoria real o principal, este espacio es un conjunto de ubicaciones físicas de memoria en el cual las instrucciones y datos de un programa deberán ser ubicados para su procesamiento. El número de direcciones en esos dos espacios no necesariamente es el mismo a pesar de que ambos empiezan con la dirección 0 y tienen direcciones consecutivas.



Los programas se refieren a los datos e instrucciones por la dirección de la memoria virtual sin conocer la ubicación física de la memoria principal.

La memoria virtual es así denominada porque representa una imagen de la memoria física. Dado que la memoria virtual no existe como una entidad física de la memoria principal, las instrucciones o datos referenciados por direcciones virtuales deben ser contenidos en alguna ubicación física de memoria para ser ejecutados.

MODO DE FUNCIONAMIENTO (Paginación – Segmentación)

Los contenidos de la memoria virtual están divididos en páginas de tamaño fijo. El programa estará completo en la memoria virtual pero en la memoria real sólo estarán algunas páginas del mismo que irán cambiando a lo largo de su ejecución. Esto se realizará en forma automática.

El espacio de direccionamiento de la memoria virtual estará contenido en dispositivos de acceso directo y corresponde a los programas que se están ejecutando.

A su vez, la memoria real también está dividida en bloques de igual tamaño que las páginas. Cuando se debe ejecutar un programa, éste es traído de la biblioteca a la memoria virtual y las instrucciones y datos del programa son transferidos entre la memoria virtual y la real de a un bloque por vez durante la ejecución del programa.

El hardware de traducción de direcciones o la función interna de mapping son los mecanismos por los cuales se pueden traducir las direcciones de memoria virtual en direcciones de memoria real durante la ejecución de las instrucciones. El sistema operativo mantiene distintas tablas que indican, entre otros datos:

El sistema operativo mantiene distintas tablas que le indican:

1. Cantidad de memoria virtual implementada.
2. Secciones que están presentes en la memoria real.
3. Direcciones indicando la ubicación en memoria real de cada una de dichas secciones.
4. Elementos de juicio para determinar que secciones se tratarán de dejar en la memoria real, y cuáles no, o eventualmente, que sección será desplazada cuando otra página de memoria virtual deba ser llevada a la memoria real.



La tabla de cada programa contiene el siguiente formato:

Nº de Página	Indicador Memoria Real	Indicador de páginas modificadas	Indicador de uso	Dirección de Memoria Real
1	1	1	X	Dirección 1
2	0	0		
3	0	0		
4	1	0	Z	Dirección 2
5	0	0		

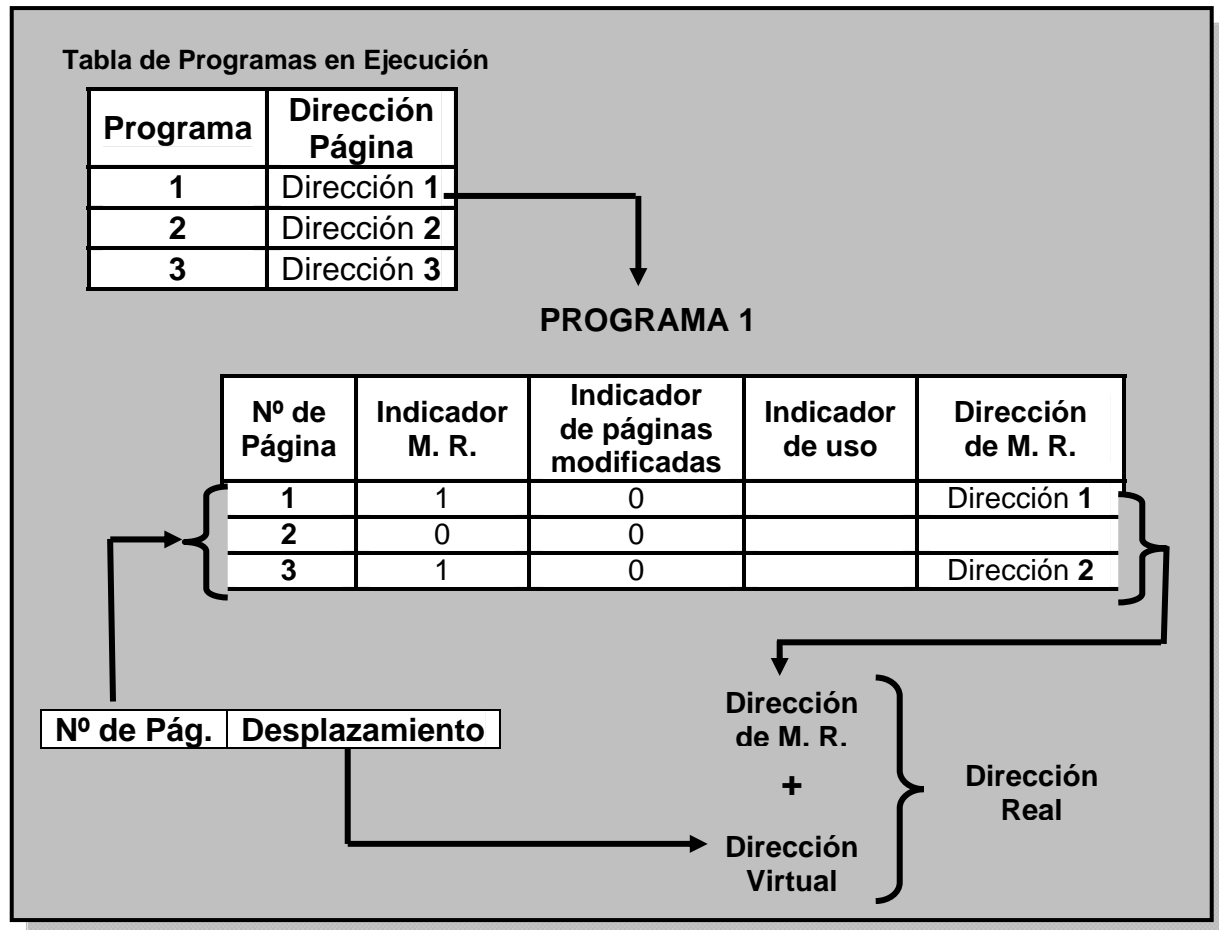
Información por columna:

- 1º. Indica el número de página del programa.
- 2º. Marca que indicará si se encuentra en memoria real (código 1) o no (código 0).
- 3º. Indicará, para las que se encuentran en memoria real, si han sido modificadas (código 1) o no (código 0).
- 4º. Le dará al sistema operativo otros elementos de juicio para decidir que página será reubicada y cuando se deba traer otra página a la memoria real.
- 5º. Brindará la dirección de memoria real de aquellas páginas que se encuentren en ella.

Al utilizar memoria virtual, las direcciones virtuales no se expresan, como habíamos visto anteriormente, con registros base y límite (desplazamiento). En realidad existen distintas variantes. La más sencilla es aquella que direcciona una página del programa y un desplazamiento dentro de la página. Si consideramos que un programa está dividido en páginas de tamaño fijo, la dirección podrá ser expresada como número de página y desplazamiento dentro de ella. El gráfico de conversión sería el siguiente:



Forma Paginada



Memoria virtual a convertir se buscará en la tabla del programa y se determinará, mediante el indicativo de presencia en memoria real, si la página referenciada se encuentra o no en memoria real. En caso afirmativo se tomará la dirección de comienzo de la página que figura en la última columna de la tabla, se le sumará el desplazamiento dentro de la página de la dirección virtual y el resultado nos dará la dirección de memoria real.

Otra forma de direccionamiento virtual consiste en subdividir la en 3 niveles:

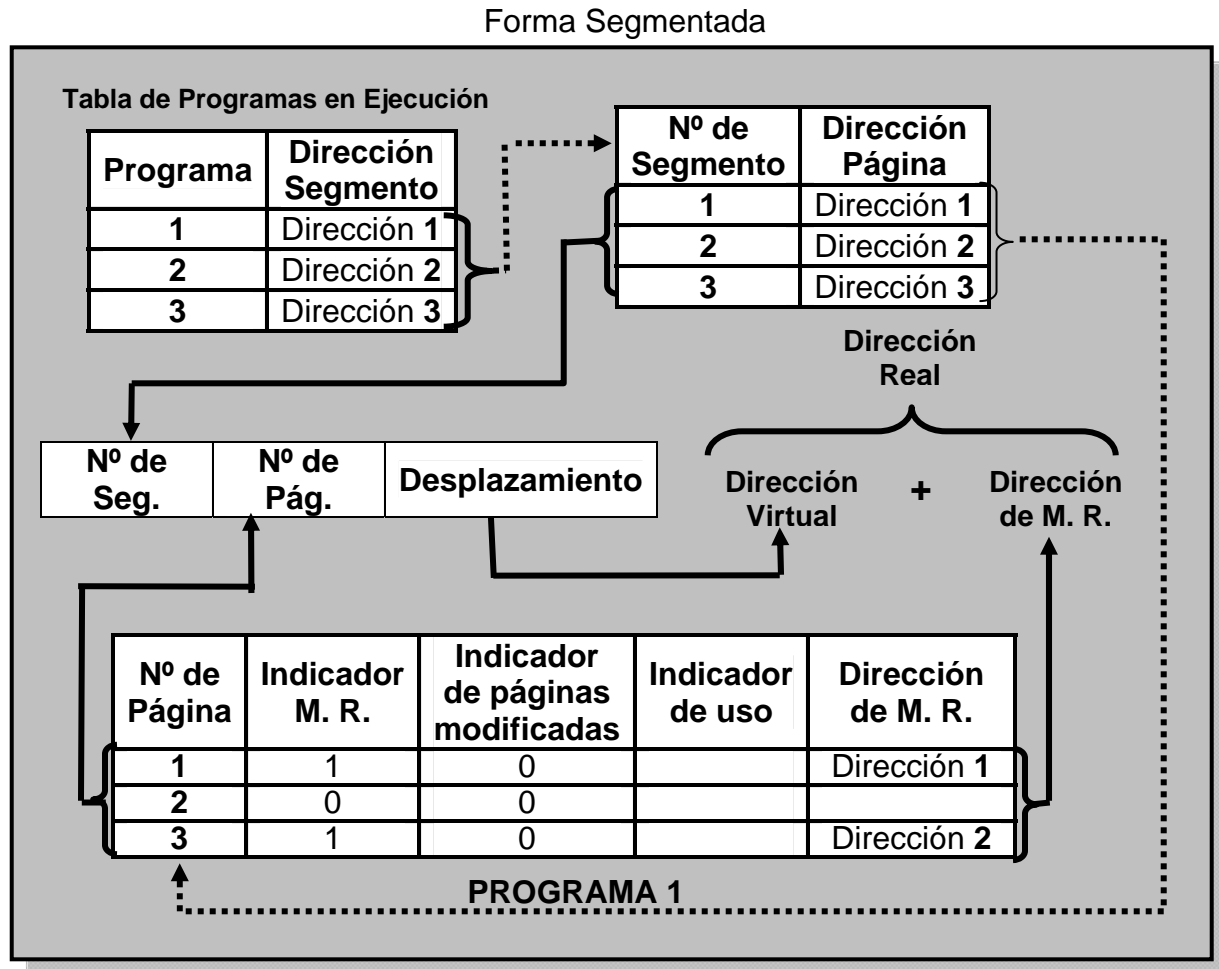
- Número de Segmento.
- Número de página dentro del segmento.
- Desplazamiento dentro de la página.

Cada segmento puede tener cómo máximo un determinado número de páginas (determinado por la cantidad de bits para expresar el número de página dentro del segmento).



En este caso la tabla de cada programa se divide en dos: una tabla de segmentos que se completa con apuntadores hacia las tablas de las páginas correspondientes a cada segmento.

Cada segmento tendrá una cantidad determinada de páginas. El gráfico de conversión sería el siguiente:



Cada vez que haya que convertir una dirección virtual en real se accederá a la tabla de programas en ejecución que indicará la dirección de comienzo de la tabla de segmentos del programa correspondiente. Con esta dirección y el número de segmento de la dirección virtual a convertir obtendremos la dirección de comienzo de la tabla de páginas del segmento. Nuevamente, con esta dirección y el número de página dentro del segmento de la dirección virtual obtendremos la dirección de comienzo de la página en memoria real (siempre que se encuentre alojada en la misma). Sumando a esta dirección el desplazamiento dentro de la página accede a la dirección real.



VENTAJAS Y DESVENTAJAS

En líneas generales las ventajas globales de la memoria virtual son:

- Con respecto a la performance del sistema: Además de la ventaja de contar con una memoria virtualmente más extensa con la memoria real, ésta se utiliza mucho más eficientemente. Este esquema permite que esa parte que no se utiliza no ocupe lugar en la memoria y en cambio quede disponible para agregar más programas en la mezcla de multiprogramación e incrementar de esta manera las ventajas que ésta ofrece.
- Con respecto al desarrollo de aplicaciones: Es indudable que al eliminar prácticamente las restricciones de memoria, permite diseñar los sistemas con mayor facilidad y tal vez en menor tiempo y a menor costo.
- Con respecto a la operación de la computadora: Permite que un equipo de una memoria real menor pueda ser usado sin dificultad como backup en caso de necesidad. También se simplificará su planificación.

Los beneficios potenciales de la memoria virtual serán alcanzados en mayor o menor grado, dependiendo básicamente de la configuración de la computadora y del tipo y complejidad de sus sistemas de aplicación.

Por otro lado, podemos señalar los siguientes inconvenientes:

- El costo asociado a la transformación de direcciones.
- Memoria adicional de requiere para almacenar las tablas descritas (memoria real de la parte residente del sistema operativo).
- Cierta desperdicio de memoria que se produce en la página de un programa (ya que difícilmente el tamaño de un programa sea múltiplo del tamaño de las páginas).
- Merma en el rendimiento de la computadora si es incorrectamente utilizada.
- El tiempo de ejecución de cada programa puede incrementarse como consecuencia de la paginación (operaciones de E/S que demorarán la ejecución del programa). Aunque esto, como en multiprogramación, cada programa puede tener mayor tiempo de ejecución logrando de esa forma un mayor aprovechamiento de la CPU.



RENDIMIENTO

Como hemos visto, memoria virtual puede ser una herramienta poderosa para incrementar la performance o rendimiento de la computadora. Sin embargo, esto dependerá de algunos factores:

- La velocidad de los dispositivos que contendrán la memoria virtual.
- La velocidad del procesador.
- El tamaño de la memoria real y virtual.
- La estructura de los programas.
- Algunas de las formas para incrementar al rendimiento del equipo son:
- Usar áreas de E/S más largas (registros de memoria intermedia).
- Aumentar la capacidad del manejo de páginas faltantes cuando la actividad de paginación causa frecuentes desactivaciones de programas.

Esto puede ser realizado:

- 1) Usando un dispositivo de mayor velocidad.
- 2) Estableciendo más dispositivos de paginación para permitir más superposiciones (intercambio) de actividades de paginación.
- 3) Reduciendo o eliminado archivos del dispositivo de paginación.
- 4) Reduciendo la actividad del canal que corresponde al dispositivo de paginación.
- 5) Aumentando, en lo posible, el tamaño de la memoria real.
 - Estructurar los nuevos programas de aplicación para que operen eficientemente con memoria virtual.
 - Incrementar el nivel de multiprogramación.