

SISTEMAS OPERATIVOS

Elementos de Informática

¿Qué es un Sistema Operativo?

El término S.O. denota aquellos módulos de programa dentro de una computadora que gobiernan el control de los recursos del equipo tales como procesador, memoria principal, memoria auxiliar, dispositivos de E/S y archivos.

-

Estos módulos tratan de optimizar la performance y simplificar el uso eficiente del sistema.

Vista abstracta de un Sistema de Cómputos - Esquema de Peterson



Evolución de los S.O.

Comienzos de la Computación

En los primeros sistemas de computación el operador debía poner en marcha la computadora en un proceso complejo.

La evolución avanza con la llegada de los dispositivos de E/S, así como de los traductores de lenguaje.

El "cargador", automatiza el proceso de cargar en memoria los programas ejecutables y datos del usuario guardados en las tarjetas perforadas y las cintas perforadas.

Evolución de los S.O.

Procesamiento en serie

En los comienzos la carga de los programas se realizaba en un proceso de ejecución en serie.

Contaban con un conjunto de rutinas de E/S estándar para usar con todos los sistemas, dada la gran cantidad de programas que usaban estos dispositivos.

- Las rutinas de E/S (drivers)
- El cargador.

Estos constituían un entorno mínimo de ejecución, es decir que ya representaban una forma rudimentaria de SO

Evolución de los S.O.

Procesamiento por Lotes

La siguiente evolución de los SO fue automatizar la secuencia de operaciones involucradas en la ejecución de un programa.

En una cinta de entrada se agrupan varios programas para que las operaciones auxiliares se realicen una sola vez así el tiempo de ocupación por programa, se reduce considerablemente.

Evolución de los S.O.

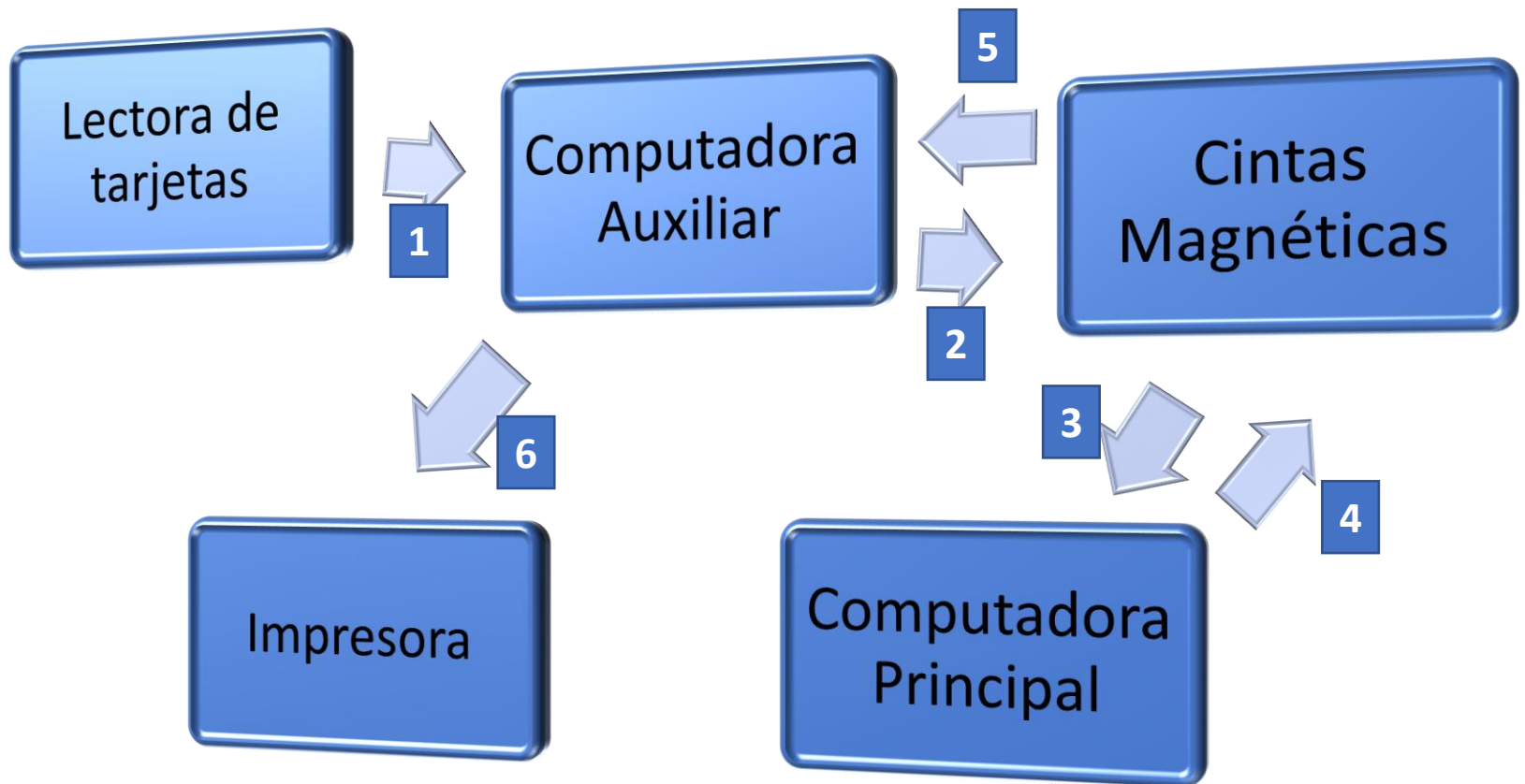
Procesamiento por Lotes

Para evitar recargar compiladores, vinculadores o cargadores, muy pronto estos programas se hicieron "residentes" , es decir, se podían correr sin que fuera necesario volver a leerlos.

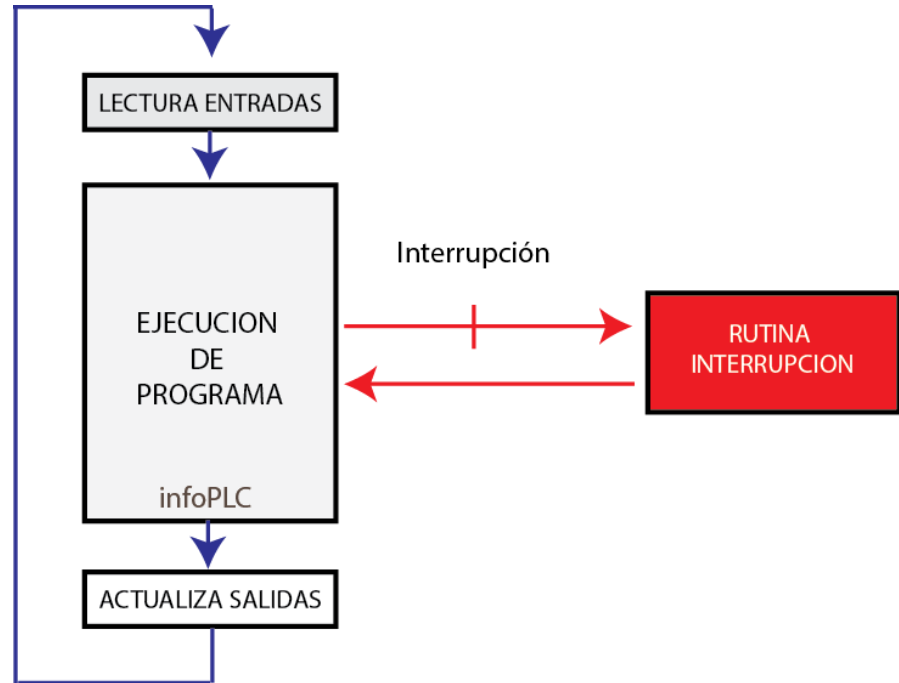
Para aprovechar esta capacidad, el operador agrupaba en lotes (batches) los programas con requisitos semejantes.

Evolución de los S.O.

Procesamiento por Lotes



El siguiente desarrollo lo introdujo la llegada de una técnica de hardware: la rutina de interrupción. Esta rutina permite conservar el estado del hardware ante la finalización o la presencia de un error durante la ejecución, permitiendo reanudar el programa interrumpido.



Evolución de los S.O.

Procesamiento por Lotes

Este concepto se utilizó para obtener un mejor uso del equipo por medio de la técnica de *multiprogramación*.

La tarea de i/o se encuentra a cargo del equipo central utilizando el monitor de SPOOL (Simultaneous Peripheral Operations On Line = Operaciones periféricas simultáneas en línea) permitiendo la concurrencia de la E/S.

Evolución de los S.O.

Multiprogramación

La multiprogramación implica que muchos programas residan en memoria al mismo tiempo.

El SO toma uno de los programas residentes en memoria y comienza a ejecutarlo.

Si eventualmente debe esperar por algo, tal como un comando del teclado, o una operación de E/S el SO simplemente toma otro trabajo y lo ejecuta.

La CPU nunca estará inactiva.

Evolución de los S.O.

Multiprogramación

Los trabajos deben estar activos en memoria para poder ser ejecutados; esto implica algún tipo de manejador de memoria.

Dado que todos los programas están listos para correr al mismo tiempo; se deberá tomar una decisión entre ellos para elegir cuál tendrá la atención de la CPU.

Este trabajo lo realiza el manejador del procesador.

Evolución de los S.O.

Sistemas de Tiempo compartido y Multiacceso

Los sistemas de tiempo compartido son una variante de la multiprogramación.

El uso del procesador en estricta secuencia entre un programa y otro, en periodos de tiempo muy pequeños, simula la ejecución de más de un programa al mismo tiempo (paralelos) permitiendo dar respuesta a más de un usuario.

Sistemas Operativos: Definición

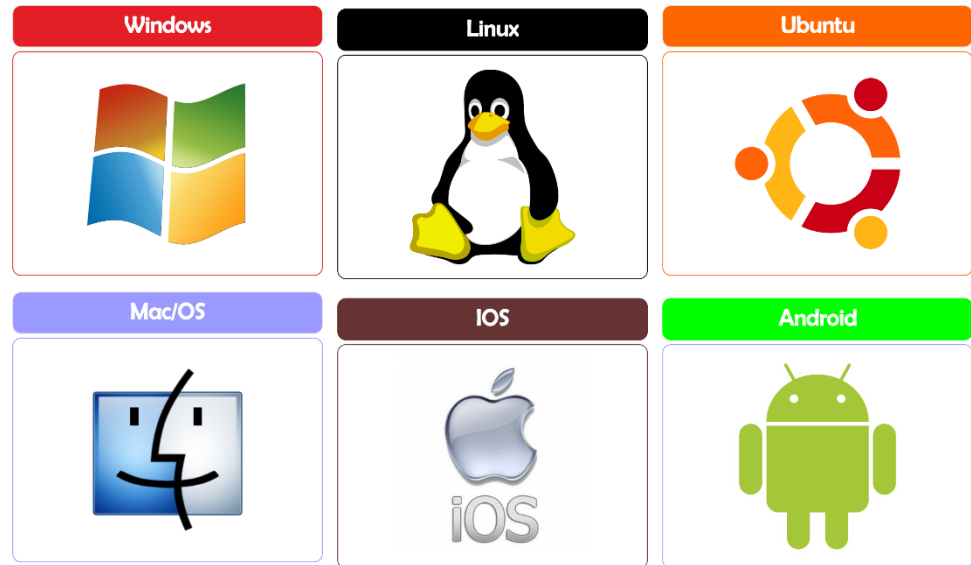
Según Milenkovic un SO se puede ver como una colección organizada de software (que extiende al hardware) y que consta de rutinas de control para operar una computadora y proporcionar un entorno para la ejecución de programas.

Para Barron, el SO es como el pegamento que mantiene juntos a todos los otros componentes del software de sistema: editores, ensambladores, compiladores, etc.

Sistemas Operativos: Definición

Se puede considerar que proporciona una infraestructura, un medio común en el cual los

diferentes programas del sistema pueden operar, incluyendo los mecanismos para comunicarse con los diferentes dispositivos.



Sistema Operativo: Misiones

1) Controlar los Recursos:

Los dispositivos que componen el hardware deben estar bajo el control del SO, deben iniciar su actividad cuando éste lo dispone, y al concluirla, éste debe ser quien reciba el mensaje y discrimine la acción a seguir.

Sistema Operativo: misiones

2) Administrar los Recursos:

El SO no sólo debe ocuparse de controlar los recursos sino también de administrarlos.

Todos los dispositivos de un sistema de cómputo están a disposición de los usuarios de un SO a través de él. El SO debe intentar maximizar el aprovechamiento de estos recursos sin provocar conflictos.

Sistema Operativo: misiones

3) Proveer un ambiente amistoso:

La interfaz con el usuario debe permitirle hacer lo que quiere de varias maneras, todas combinaciones de funciones elementales sencillas y poderosas.

También debe informar los problemas que encuentra de una manera simple y clara.

Sistema Operativo: funciones

1) Ejecución de Programas: Este servicio es básico para la operación de la computadora. La función principal del SO es la de brindar un ambiente de ejecución de programas. Debe permitir: almacenar, cargar, ejecutar e interrumpir programas de usuarios.



Sistema Operativo: funciones

2) Operaciones de E/S: Es el SO quien debe realizar las operaciones en cuestión, pero eso le implica una administración de las solicitudes que realizan los usuarios.



Sistema Operativo: funciones

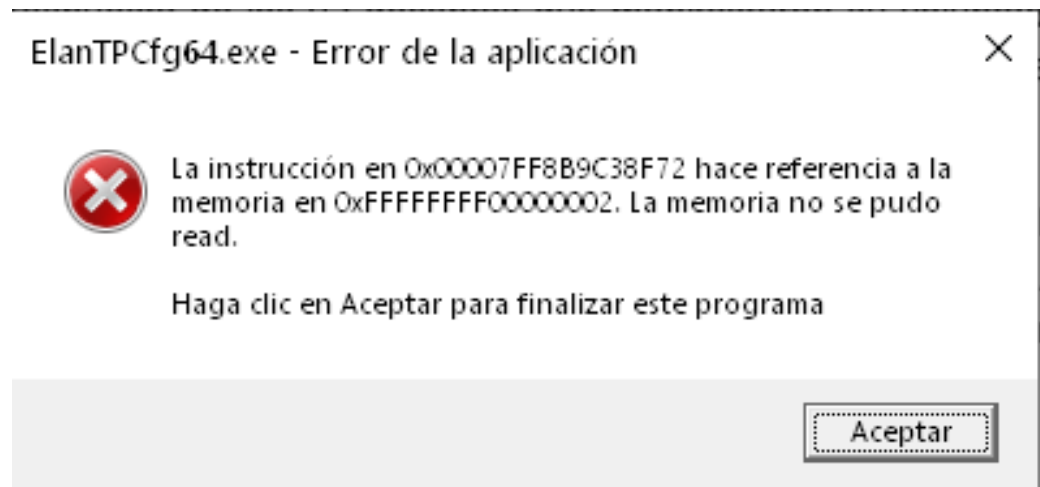
3) Manipulación de archivos: A través del SO, se canalizan servicios esenciales de almacenamiento, recuperación, memoria secundaria y apoyo a la ejecución de programas.



Sistema Operativo: funciones

4) Detección de errores: los SO deben proveer funciones de atención de errores que puedan tener lugar durante la ejecución de programas de usuario, evitando su propagación. El error puede ocurrir en la CPU, en el hardware, en los dispositivos de E/S, etc.

Para cada tipo de error, el SO toma la acción apropiada para asegurar el correcto y consistente cómputo.



Sistema Operativo: funciones

5) Adjudicación de Recursos : Un sistema operativo debe distribuir sus recursos entre los usuarios según su necesidad y las posibilidades reales que haya; sin permitir situaciones de “abrazo mortal” entre los programas que compiten por los mismos recursos.



Sistema Operativo: funciones

6) Contabilidad: Una computadora es un instrumento destinado a brindar servicios de procesamiento de datos, es de interés común saber cómo y por quiénes se utilizan los recursos del equipo.

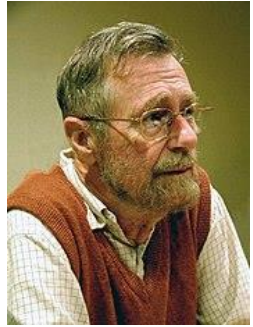


Sistema Operativo: funciones

7) Protección: El SO debe cuidar que las funciones no perjudiquen a otros usuarios o al propio SO. En combinación con el hardware, el SO protege la operación del equipo, previniendo la invasión de sectores de memoria no accesibles, la ejecución de instrucciones privilegiadas o el uso de recursos para los que no se cuenta con la debida autorización.

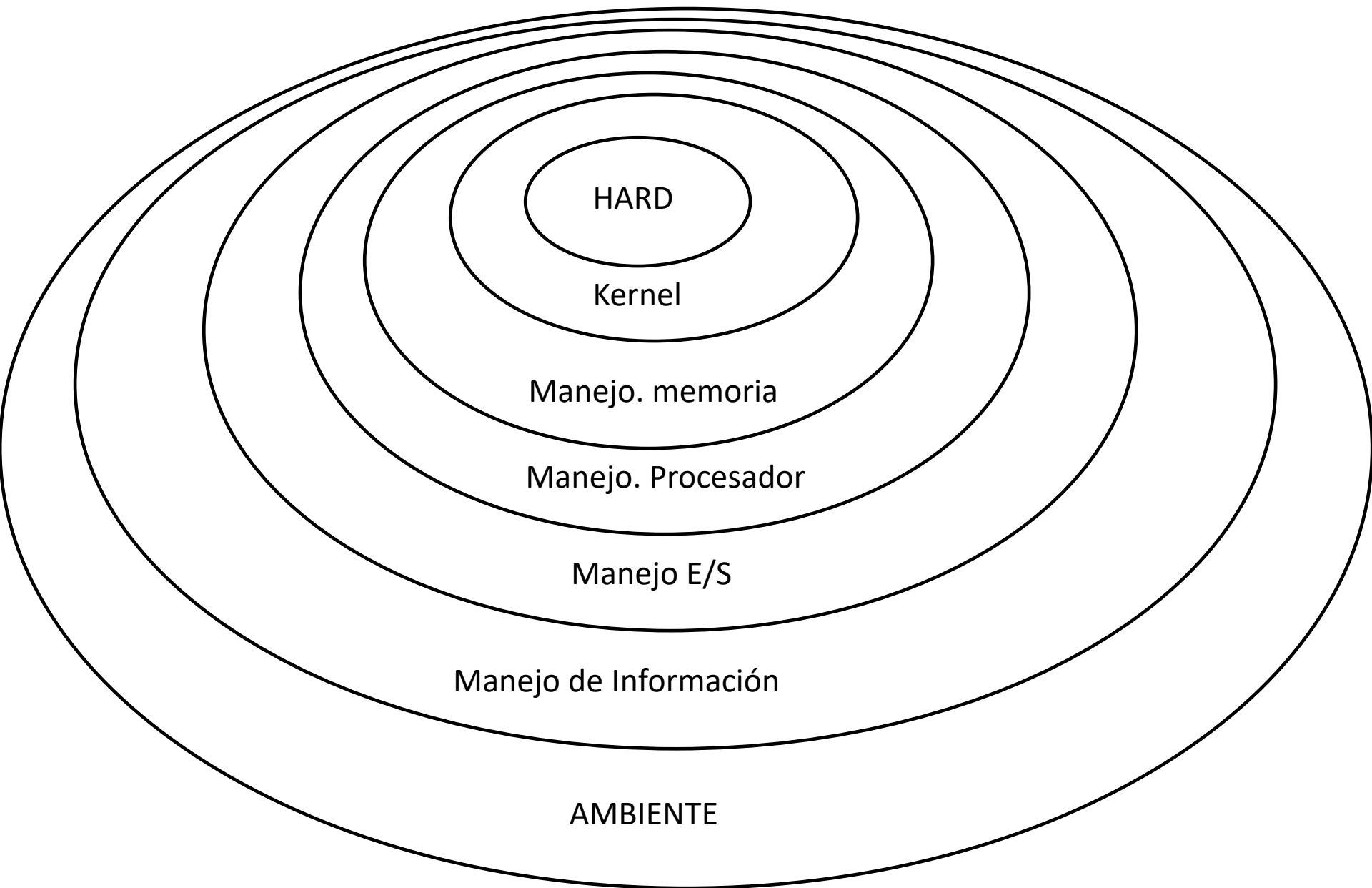


Modelo de Estudio para los Sistemas Operativos



Según Dijkstra la construcción del SO presenta un diseño jerárquico en 5 niveles (capas de creciente nivel de abstracción). Cada nivel representa una funcionalidad distinta, permitiendo la depuración del sistema completo en sucesivas aproximaciones.

Siguiendo a Dijkstra y su “modelo de la cebolla” estudiaremos los SO con funciones jerárquicas que van desde los niveles muy cercanos a la máquina misma hasta niveles más “virtuales”.



Núcleo o Kernel del S.O

El núcleo del SO es la capa más interna que corre directamente sobre el hardware.

Es una cantidad mínima de código esencial para su funcionamiento y es usado intensivamente por todos los programas situados en niveles superiores.

Las funciones del núcleo consisten en tomar el control del procesador y determinar cuándo y cómo se lo va a repartir entre diferentes usuarios.

Por ejemplo, el problema de la concurrencia entre procesos, tiene que resolverse al nivel más cercano al núcleo del SO.

Manejo de Memoria

Un proceso cualquiera puede estar en uno de varios estados:

Activo

En ejecución o

Residente en disco magnético.

La función del manejador de memoria consiste en mantener un espacio en la misma para todos los procesos activos dentro del sistema.

Si hay otros programas que deseen acceder a la memoria principal, se les mantiene normalmente en espera en un área de almacenamiento de reserva.

Manejo de Memoria

Una parte del almacenamiento principal se usa permanentemente para contener la porción residente del SO, el núcleo.

El núcleo debe funcionar correctamente para mantener un aislamiento de c/programa de usuario que se está ejecutando y evitar él mismo ser adulterado.

Existen varias maneras de manejo de memoria en un sistema de cómputo.

Manejo del Procesador

Este programa del sistema (también conocido como Manejador de Alto Nivel, Scheduler) tiene como función determinar en qué orden y con qué criterios se les va a dar atención a los diversos usuarios de un sistema de cómputo.

Manejo de Entradas y Salidas

El manejador de E / S tiene como función principal atender los pedidos que los procesos en ejecución hacen sobre las unidades periféricas.

Esta atención requiere, la mayoría de las veces, una traducción lógica y física entre las diversas unidades involucradas.

La parte física logra que aparatos diferentes entre sí puedan comunicarse aunque manejen códigos internos distintos.

Manejo de Información

Esta capa del SO se encarga de las transferencias físicas de información entre las unidades de E / S y los procesos en ejecución.

Sus funciones son permitir a los usuarios el manejo libre y simbólico de prácticamente cualquier cantidad de información que desee almacenar, leer, imprimir, alterar o desechar.

EL sistema puede almacenar la información por plazos indefinidos y recuperarla en cualquier momento, a la vez que maneja criterios de seguridad de acceso y protección.

Gestión de Memoria

La gestión de memoria de un SO *asigna y libera* porciones de memoria principal en respuesta a las peticiones de los usuarios y de otros módulos del SO de acuerdo con los objetivos de la gestión de los recursos en un sistema particular.

Los procesos están creados y cargados en memoria principal, en respuesta a algún **tipo de planificación**.

Se les asigna una cantidad de memoria en un instante dado, y la memoria se libera cuando los objetos residentes terminan.

Gestión de Memoria

El objetivo de la gestión de memoria es suministrar un uso eficiente de la memoria, minimizando la cantidad de memoria desaprovechada.

Además, el gestor de memoria debe suministrar protección aislando los diferentes espacios de direcciones y facilitando la cooperación entre los procesos, permitiendo el acceso a código y datos compartidos.

Técnicas de Manejo de Memoria

1 - Asignación Única Contigua

Es un esquema muy simple de asignación de memoria, asociado con pequeñas computadoras. En dichos sistemas no es posible la multiprogramación.

La memoria está dividida conceptualmente en 3 regiones contiguas:

- Área permanente para el SO.
- El espacio remanente está disponible para la tarea simple que está siendo procesada.
- Memoria disponible.

Asignación Única Contigua



TODAS LAS DIRECCIONES DESARROLLADAS POR EL PROGRAMA DEL USUARIO SON VERIFICADAS PARA ASEGURAR QUE NO SON MENORES QUE "A".

Asignación Única Contigua

Toda la memoria es asignada a cada proceso.

Su única ventaja es la simplicidad, no requiere mucha experiencia para entender su manejo.

En contraposición, sus desventajas son:

- ✓ pobre utilización de memoria y procesador
- ✓ Los trabajos del usuario están limitados al tamaño de memoria disponible.

Manejo de Memoria por particiones

Consiste en dividir la memoria en varias secciones fijas y asignar a c/u de ellas un usuario o proceso activo.

Un problema a resolver es asegurar que ningún usuario intervenga en el área de memoria asignada a otro. Para lo cual se utilizan interrupciones de protección.

Manejo de Memoria por particiones

El sistema puede ser de particiones estáticas o dinámicas (dinámicas significa que las particiones son creadas durante la ejecución del proceso, de modo de hacer coincidir el tamaño de las particiones con el de los procesos).

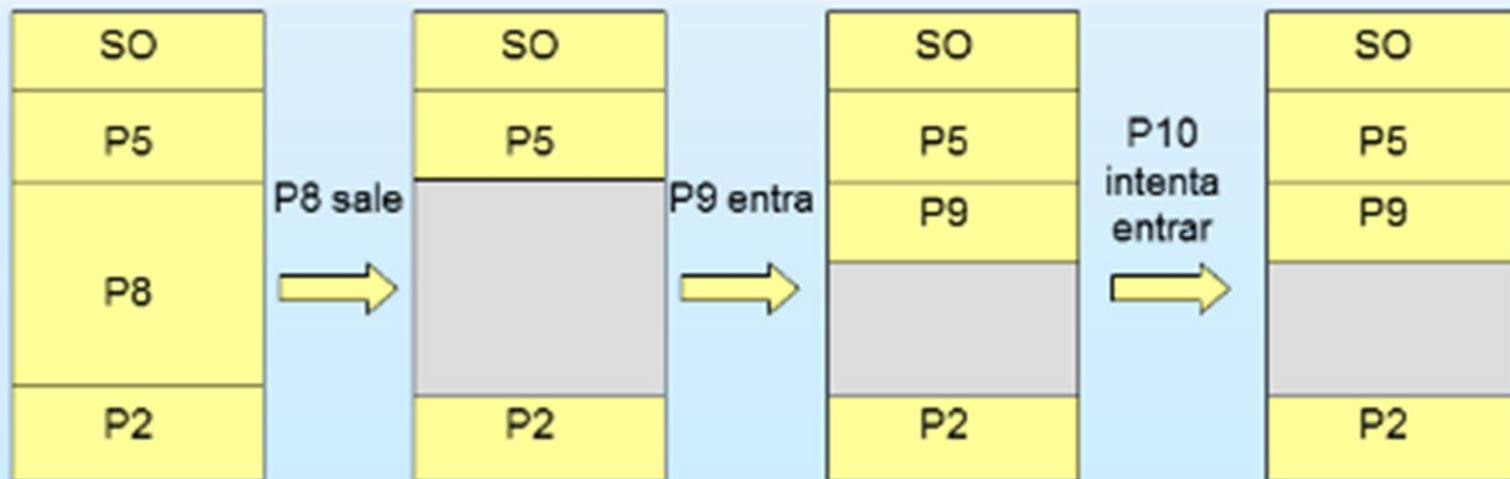
En las estáticas, toda la memoria es asignada antes de la ejecución de los procesos.

El manejo de memoria consistirá entonces, en llevar un control de qué particiones están asignadas a cuáles procesos.

Manejo de Memoria por particiones

Ventaja: permite la multiprogramación.

Desventaja: deja lugares libres en la memoria que, al ser de tamaño fijo, no pueden ser utilizadas más que por procesos de longitud igual o menor.



Manejo por particiones relocizables

Una solución al problema de fragmentación (expuesto previamente) es combinar periódicamente todas las áreas libres en un área contigua. Esto puede hacerse moviendo los contenidos de las particiones ocupadas, de modo de que se vuelvan contiguas. Este proceso se llama “compactación”.

Manejo por particiones relocizables

Las celdas en realidad no se mueven, sino que sus contenidos se copian de un lugar a otro. Este esquema es más flexible que el anterior, aunque más costoso, ya que mover una partición implica un proceso complejo de *relocalización*.

Ventaja

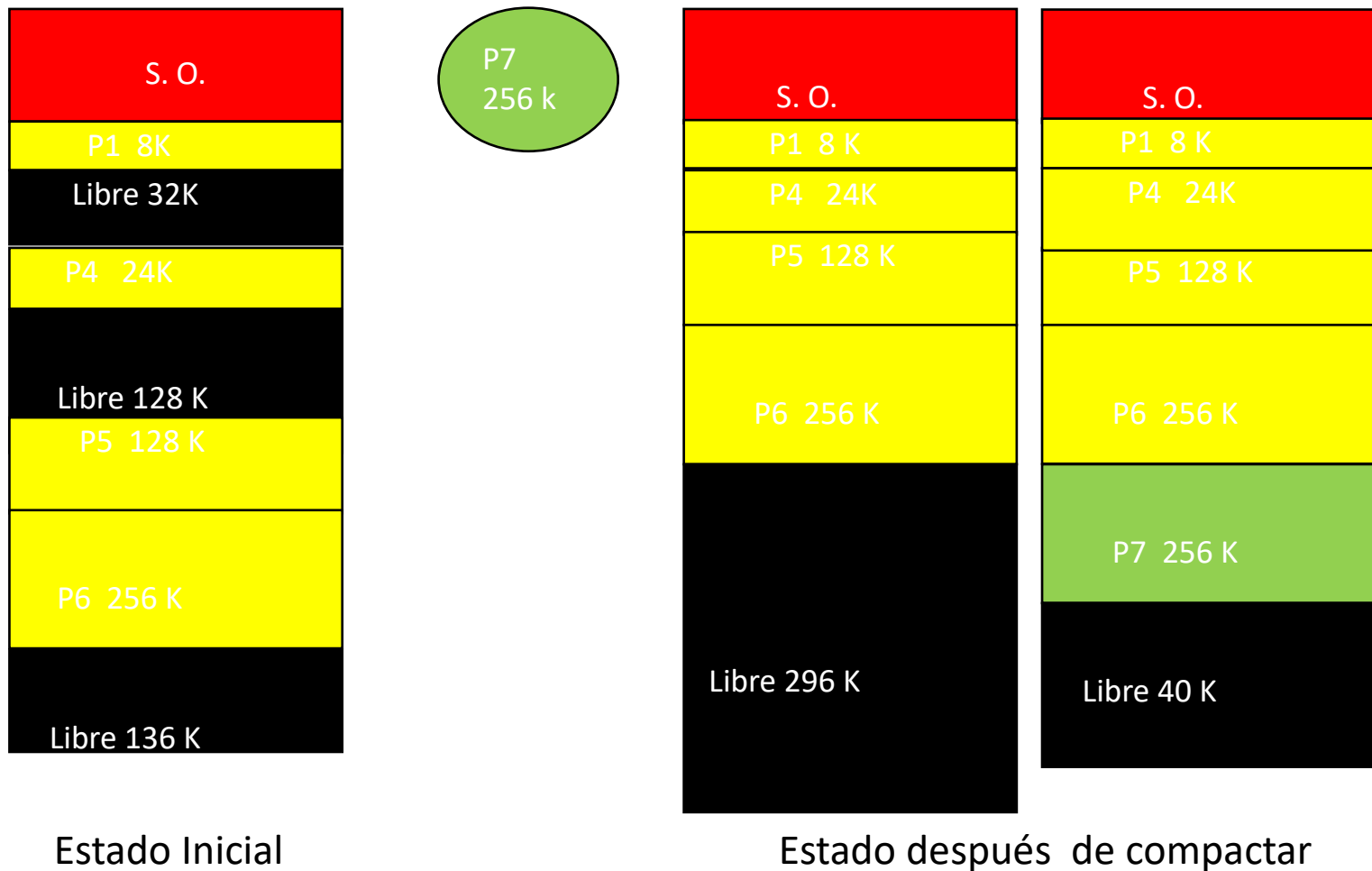
Elimina el problema de fragmentación y hace posible ocupar más particiones. Permite un mayor grado de multiprogramación.

Manejo por particiones relocizables

Desventajas:

- a. El hardware de relocización incrementa el costo de la computadora y puede reducir un poco la velocidad.
- b. El tiempo de compactación puede ser considerable.
- c. Alguna memoria puede seguir siendo inutilizada porque, aunque esté compactada, el área libre puede ser menor que el tamaño de partición necesitado.
- d. EL tamaño de particiones de los procesos está limitado al tamaño de la memoria física.

Manejo por particiones relocables



Manejo de Memoria por Paginación

Debido a los costos que representaba la relocalización, surge un manejo más ágil y eficiente, llamado “paginación”.

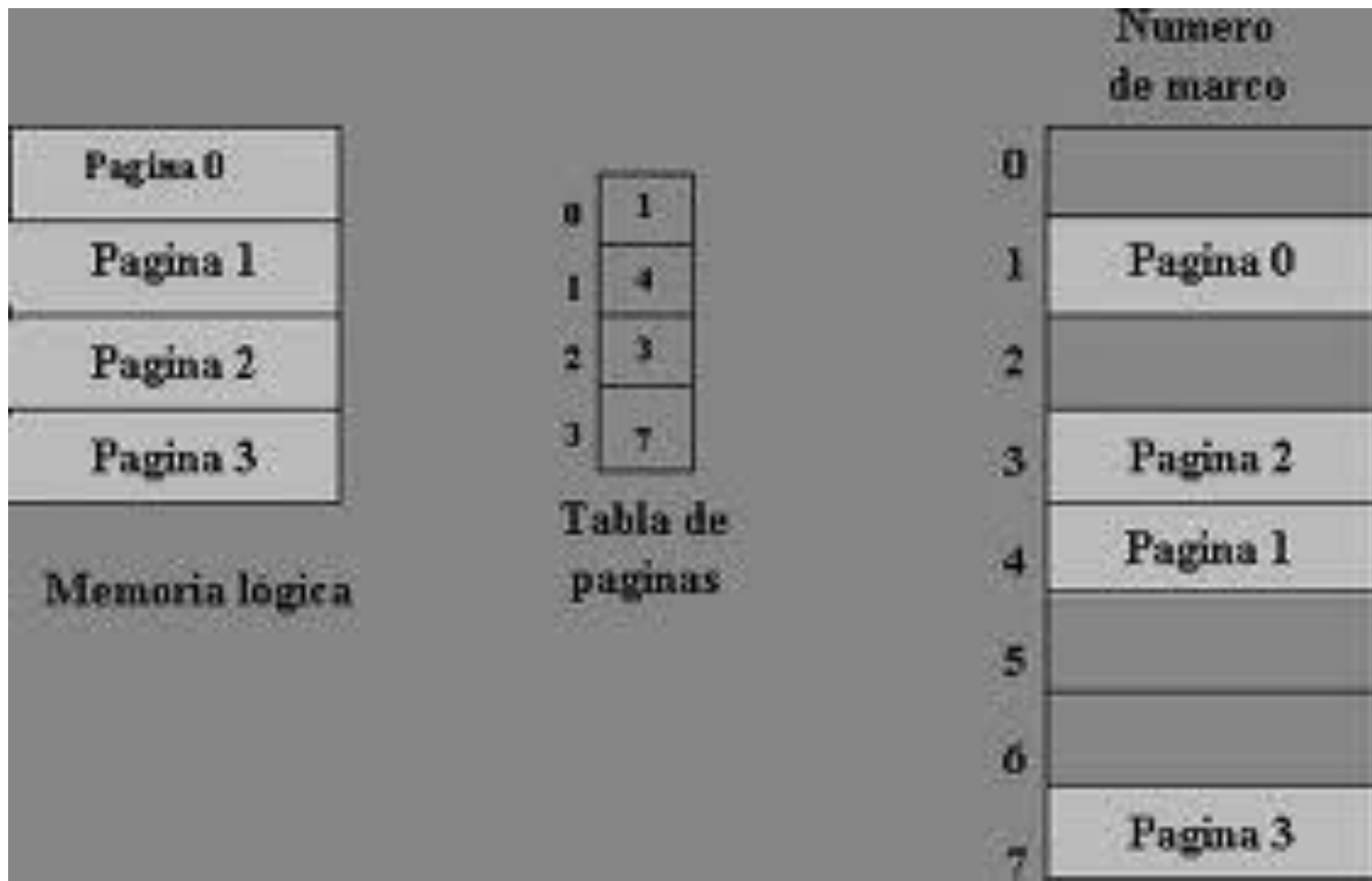
Consiste en dividir los procesos en fragmentos de longitud fija llamados “páginas lógicas”, que se almacenan en áreas de igual tamaño en memoria , llamados “bloques o marcos”.

Manejo de Memoria por Paginación

Ventajas: El esquema de memoria paginado elimina la fragmentación y hace posible acomodar el espacio de direcciones para más procesos simultáneamente. Permite mayor grado de multiprogramación.

Desventajas: El hardware requerido usualmente incrementa el costo de la computadora y al mismo tiempo es más lento. Se debe usar memoria para tablas de ubicación. Debe ocuparse tiempo del procesador para mantener y actualizar estas tablas. Como en los casos anteriores, el espacio de direcciones de procesos, está limitado al tamaño de la memoria física.

Manejo de Memoria por Paginación

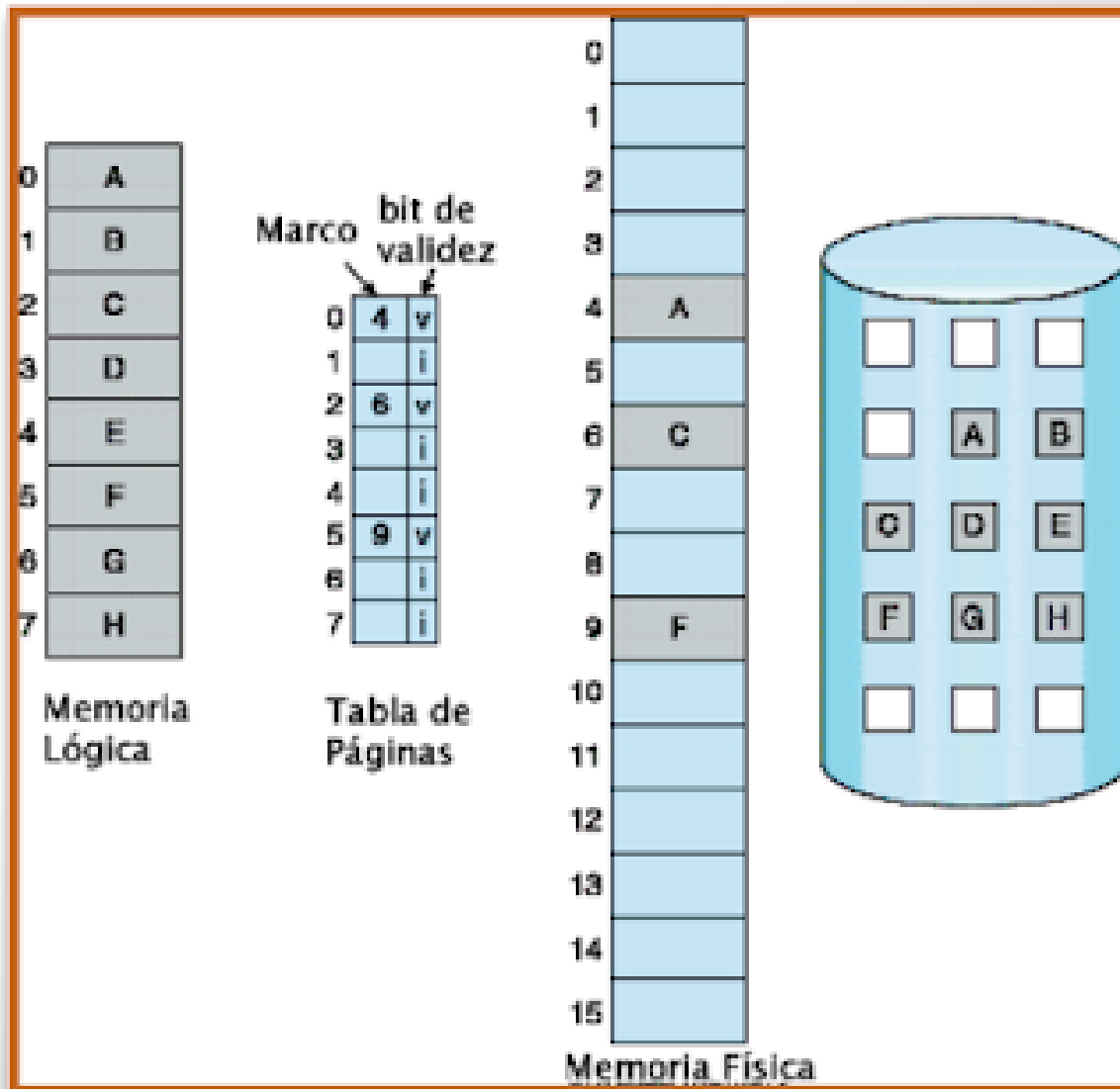


Paginación por demanda

En los métodos anteriores, un proceso no podía correrse a menos que hubiera suficiente memoria disponible para cargar su espacio de direcciones completo. Esta restricción, a menudo provocaba la existencia de áreas libres en memoria, aún cuando había procesos esperando para ser cargados y ejecutados.

Un método para trabajar con una memoria muy grande, es usar el SO para simular que se cuenta con ella; como esta memoria es sólo una ilusión se le denomina “memoria virtual” .

Paginación por demanda



Paginación por demanda

Ventajas: Se elimina la fragmentación y no es necesaria la compactación, utiliza memoria virtual, realiza un uso más eficiente de la memoria y permite un grado de multiprogramación no limitado por el tamaño de la memoria física.

Desventajas: Costo de hard, complejidad, tiempos grandes del procesador atendiendo el problema de “page swapping”, se requieren más tablas para el manejo de las interrupciones por página.

Segmentación

Un segmento puede definirse como un grupo lógico de información, tal como subrutinas, arrays o áreas de datos. El espacio de direcciones de cada proceso consiste en una colección de segmentos. La segmentación es una técnica de manejo de esos segmentos.

En un ambiente de segmentación, todas las referencias al espacio de direcciones requieren dos componentes:

- 1) un especificador de segmento.
- 2) la locación dentro del segmento.

Segmentación

La mayor diferencia entre un segmento y una página, es que un segmento es una unidad “lógica” de información, visible al programa del usuario y es de tamaño arbitrario. Una página es una unidad física de información invisible al programa del usuario y es de tamaño fijo, por ejemplo 4K.

Ventajas:

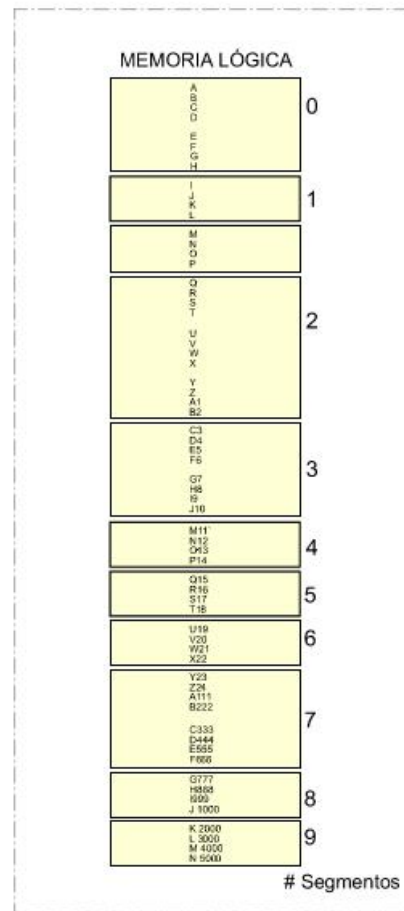
- 1) Elimina fragmentación.
- 2) Provee Memoria Virtual: guardando sólo los segmentos activos usados.
- 3) Permite segmentos que crecen dinámicamente.
- 4) Permite compartir segmentos.

Segmentación

Desventajas:

- Existe dificultad en el manejo de segmentos de tamaño variable en memoria secundaria.
- El tamaño máximo de un segmento está limitado por el tamaño de la memoria principal. El costo de hard y la complejidad siguen aumentando.

Segmentación



Segmento	Dir física
0	1700
1	1600
3	1400
6	1200
8	1300
2	1900

Tabla de Descriptores de Segmento

