

GESTIÓN DE MEMORIA

OBJETIVOS

- Organización y administración: Explorar las formas de organizar la memoria.
- Técnicas de gestión: Incluir paginación y segmentación.
- Compatibilidad con procesadores: Descripción de los procesadores Intel Pentium y sus métodos de segmentación.

FUNDAMENTOS DE LA MEMORIA

La memoria es un componente clave en los sistemas de cómputo, ya que permite el almacenamiento temporal de datos e instrucciones que la CPU necesita para ejecutar programas. Su gestión adecuada es fundamental para garantizar que los procesos puedan acceder a los recursos necesarios sin demoras, maximizando el rendimiento del sistema.

Memoria como recurso crítico:

- La memoria es esencial para el funcionamiento del sistema, ya que almacena las instrucciones y datos de los programas.

Acceso a memoria:

- La CPU accede directamente a la memoria principal para ejecutar instrucciones, lo que permite el procesamiento de tareas.

Velocidad y caché:

- Para resolver la diferencia de velocidad entre la CPU y la memoria, se utiliza una memoria caché que acelera el acceso a los datos.

HARDWARE BÁSICO

El hardware de memoria incluye componentes físicos como la memoria principal y registros en la CPU, que permiten el almacenamiento y acceso rápido a los datos necesarios para ejecutar instrucciones.

Estructura de memoria:

- La memoria principal y los registros permiten que la CPU acceda rápidamente a los datos.
Problemas de velocidad y caché:
- La diferencia de velocidad entre la CPU y la memoria se resuelve usando memoria caché, que almacena datos frecuentes para un acceso más rápido.

GESTIÓN DE MEMORIA

PROTECCIÓN DEL ESPACIO DE MEMORIA

La protección de memoria es crucial para evitar que los procesos interfieran entre sí, garantizando la seguridad y estabilidad del sistema operativo.

Registro base y límite:

- Definen el rango de direcciones que un proceso puede usar, evitando accesos no autorizados.

Prevención de errores:

- El sistema operativo restringe el acceso a la memoria para proteger los datos y prevenir errores críticos.

REASIGNACIÓN DE DIRECCIONES

La reasignación de direcciones permite que el sistema operativo maneje eficientemente la ubicación de programas en memoria, facilitando el uso dinámico de los recursos.

Ubicación de programas:

- Los programas se cargan en cualquier parte de la memoria disponible, sin necesidad de ubicarse en direcciones específicas.

Cola de entrada:

- Los programas que esperan ser cargados en memoria forman una cola, lo que permite un uso eficiente del espacio de memoria.

CARGA DINÁMICA

La carga dinámica permite cargar solo las partes necesarias de un programa en el momento en que se requieren, reduciendo el uso de memoria y mejorando el rendimiento.

- Definición: Método que permite cargar solo las partes de un programa cuando son necesarias. Esto ayuda a optimizar el uso de la memoria.
- Ventaja: Reduce la memoria ocupada por el programa, ya que las rutinas no usadas no se cargan hasta ser invocadas.
- Aplicación: Común en sistemas donde es crucial la eficiencia en el uso de memoria.

GESTIÓN DE MEMORIA

MONTAJE DINÁMICO Y BIBLIOTECAS COMPARTIDAS

El montaje dinámico carga bibliotecas solo cuando se necesitan, ahorrando memoria y permitiendo que las actualizaciones de bibliotecas se reflejen en los programas sin necesidad de recompilarlos.

- **Montaje Dinámico:** Consiste en cargar en tiempo de ejecución las bibliotecas necesarias. Se utiliza un "stub" para identificar y cargar la rutina cuando se requiere.
- **Beneficios:** Permite la actualización de bibliotecas sin necesidad de recompilar los programas que las usan.
- **Consideraciones:** Evita duplicación de código en memoria y facilita la reutilización y actualización de bibliotecas.

ESPACIOS DE DIRECCIONES LÓGICO Y FÍSICO

Las direcciones lógicas (usadas por programas) y las físicas (ubicaciones reales en memoria) están separadas para mejorar la seguridad y eficiencia, permitiendo que el sistema administre mejor la memoria.

- **Dirección Lógica:** Generada por la CPU y vista por el programa de usuario. Se le llama también "dirección virtual".
- **Dirección Física:** Es la dirección que usa la unidad de memoria para acceder a una posición física en la memoria.
- **Reasignación de Direcciones:** El esquema de reasignación de direcciones permite que las direcciones lógicas y físicas difieran para optimizar la gestión de la memoria.
- **Unidad de Gestión de Memoria (MMU):** Hardware que ayuda a transformar direcciones lógicas a físicas en tiempo de ejecución.

GESTIÓN DE MEMORIA

PARTICIÓN ESTÁTICA

Primer Ajuste

Asigna el primer hueco que tenga tamaño suficiente.

Mejor Ajuste

Asigna el hueco más pequeño que tenga tamaño suficiente.

Peor Ajuste

Asigna el hueco más grande.

Fragmentación:

Paginación: Produce fragmentación interna debido a que los bloques fijos (páginas) pueden no usarse completamente.
Segmentación: Produce fragmentación externa, ya que los segmentos tienen tamaños variables y pueden dejar espacios libres en la memoria física.

Espacios de Direcciones:

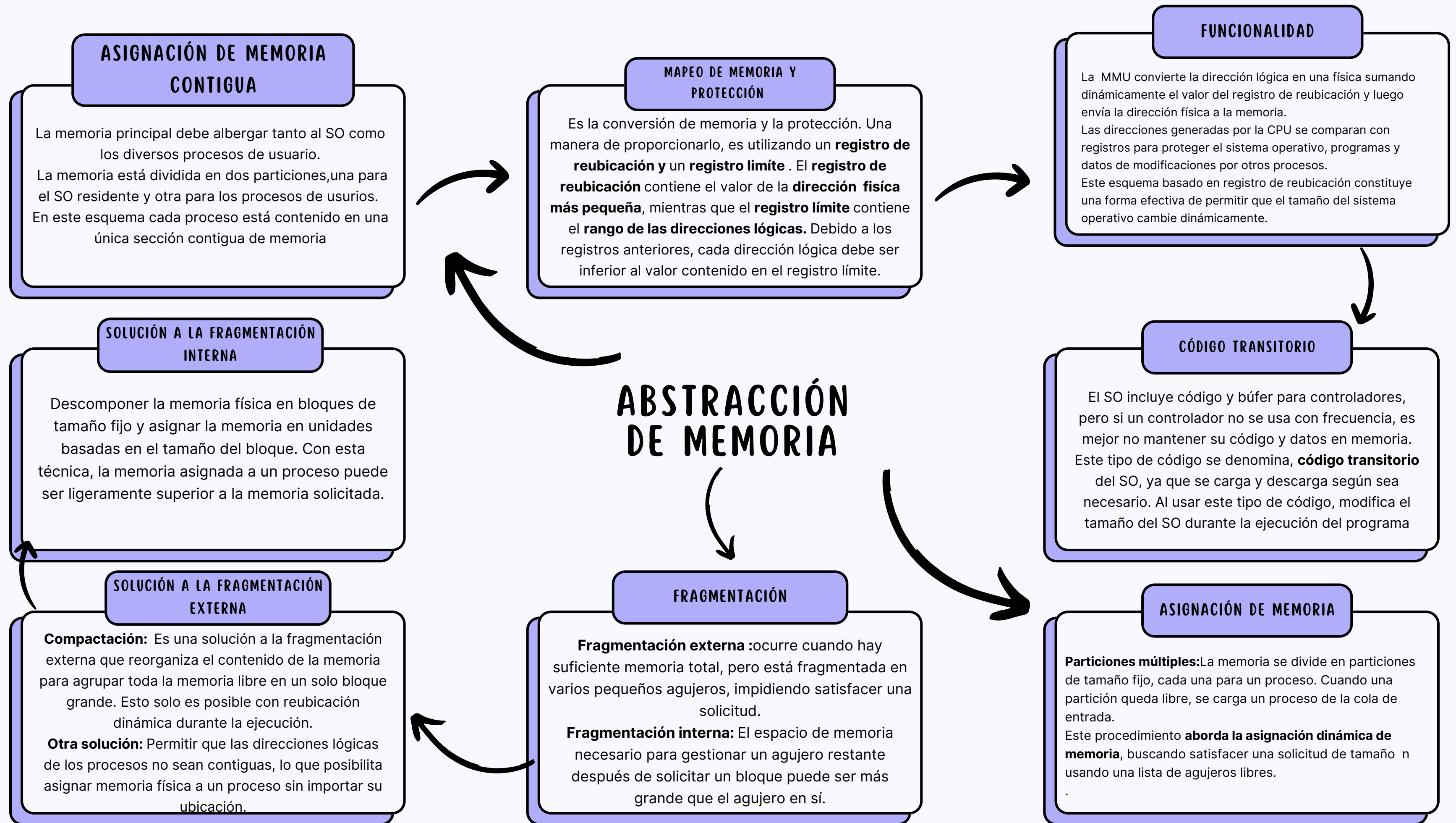
Paginación: Existe un solo espacio lineal de direcciones.
Segmentación: Hay múltiples espacios de direcciones definidos por los segmentos.

Protección Código \neq Datos:

Paginación: No ofrece una protección directa entre código y datos.
Segmentación: Sí permite proteger código y datos, ya que están en segmentos separados.

Objetivo:

Paginación: Crear un espacio lineal de direcciones para simplificar la administración de memoria.
Segmentación: Proporcionar una visión más cercana al usuario, que incluya compartición y protección.



INTERCAMBIO

Consiste en traer a la memoria cada proceso en su totalidad, ejecutarlo durante un tiempo, y después colocarlo en el disco.

Todos los procesos deben estar cargados en memoria para ser ejecutados, aunque estos pueden ser intercambiados temporalmente al sacarlos de la memoria y alojarlos en un almacen de respaldo.

ALMACÉN DE RESPALDO

El almacén de respaldo debe ser un disco rápido y lo suficientemente grande como para almacenar copias de todas las imágenes de memoria para todos los usuarios, proporcionando un acceso directo a estas. El sistema mantiene una cola de procesos preparados, formada por todos los procesos cuyas imágenes se encuentran en el almacén o en la memoria y están listos para ejecutarse.

FACTORES DE RESTRICCIÓN

- El proceso al cambiar debe estar inactivo.
- El usuario debe informar al sistema de todo cambio acerca de los requisitos de memoria.
- El tiempo de ejecución de cada proceso debe ser largo en relación con el tiempo de intercambio.
- Rapidez y tamaño adecuados del disco del almacén de respaldo.

ADMINISTRACIÓN DE MEMORIA CON MAPA DE BITS

Con un mapa de bits, la memoria se divide en unidades de asignación. A cada una de ellas corresponde un bit del mapa de bits; este es 0 si la unidad esta libre y es 1 si se encuentra ocupada (o viceversa). Cuanto menor sea el tamaño de la unidad de asignacion, mayor sera el mapa de bits.

Un tamaño excesivo origina desperdicio de memoria en la última unidad si el tamaño del proceso no es multiplo exacto de la unidad de asignación. El tamaño del mapa de bits depende del tamaño de memoria y del tamaño de la unidad de asignación, por lo que al traer un proceso de k unidades, el administrador de memoria debe buscar en el mapa una serie de k bits consecutivos con valor 0.

ABSTRACCIÓN DE MEMORIA

ADMINISTRACIÓN DE MEMORIA CON LISTAS ENLAZADAS

Consiste en mantener una lista enlazada de segmentos de memoria libres y asignados; un segmento es un proceso o bien un agujero entre dos procesos. Si los procesos y agujeros se mantienen en una lista ordenada por dirección, se pueden usar algunos algoritmos para asignar memoria a un proceso recién creado o cargado a la misma.

ALGORITMOS DE ASIGNACIÓN

Mejor ajuste. Examina toda la lista y toma el agujero más pequeño en el que pueda alojarse un proceso. Trata de encontrar un agujero con un tamaño cercano al necesario.

Peor ajuste. Este algoritmo busca tomar siempre el agujero más grande disponible, de modo que se genere la mayor fragmentación de memoria posible.

Primer ajuste. El administrador examina la lista de segmentos hasta encontrar un agujero con el tamaño suficiente, reanudando la búsqueda desde el inicio.

Siguiente ajuste. Funciona como el algoritmo de primer ajuste, con la diferencia de que en este caso se toma nota de la posición en la que se encuentra un agujero apropiado, reanudando la búsqueda desde este punto.