- 1) Explique a qué hacen referencia los siguientes términos:
  - Dirección lógica o virtual
  - Dirección física

#### Dirección lógica

- Es una dirección generada por el programa en ejecución.
- Es independiente del hardware y se utiliza en el nivel de software.
- Es manejada por la Unidad de Gestión de Memoria (MMU) y corresponde a un espacio de memoria que el programa recibe como continuo.
- Este tipo de dirección permite a los programas operar como si tuvieran acceso a toda la memoria disponible, aunque físicamente el sistema tenga limitaciones
- Un ejemplo práctico de uso es cuando varios procesos comparten un mismo recurso físico, pero cada uno cree tener su propio espacio privado de memoria.

#### Dirección física

- Es la ubicación real en la memoria RAM del sistema.
- Es administrada por el hardware del sistema y la MMU se encarga de traducir las direcciones lógicas a direcciones físicas.
- Solo es conocida directamente por el procesador y otros componentes del sistema.
- El sistema operativo se asegura de que las aplicaciones nunca trabajen directamente con estas direcciones para garantizar protección y aislamiento entre procesos.

La conversión de dirección lógica a física se realiza a través de una combinación de hardware (MMU) y software (tablas de páginas o segmentación).

- Permite abstraer el hardware y brindar flexibilidad en la administración de memoria.
- Garantiza seguridad al aislar procesos entre sí.
- Facilita características como la memoria virtual, que permite que programas utilicen más memoria de la que realmente está disponible en la RAM al usar espacio en disco.

2) En la técnica de Particiones Múltiples, la memoria es dividida en varias particiones y los procesos son ubicados en estas, siempre que el tamaño sea del mismo, menor o igual que el tamaño de la partición.

Al trabajar con particiones se pueden considerar 2 métodos (independientes entre sí)

- Particiones Fijas
- Particiones Dinámicas
- a) Explique cómo trabajan estos 2 métodos. Cite diferencias, ventajas y desventajas.
- b) ¿Qué información debe disponer el SO para poder administrar la memoria con estos métodos?
- c) Realice un gráfico indicando cómo realiza el SO la transformación de direcciones lógicas a direcciones físicas.

#### a) Métodos de particiones: Fijas vs Dinámicas

#### Particiones Fijas

- La memoria se divide en bloques de tamaño fijo al momento de la inicialización del sistema.
- Cada partición puede contener un solo proceso.
- Los tamaños de las particiones no cambian durante la ejecución del sistema.

#### **Ventajas**

- Fácil de implementar, ya que la división es estática.
- Menor sobrecarga del sistema operativo porque no necesita reorganizar la memoria frecuentemente.

#### Desventajas

- Fragmentación interna: Si un proceso es más pequeño que la partición asignada, el espacio sobrante dentro de la partición no se utiliza.
- Ineficiencia: Procesos grandes que no caben en ninguna partición no pueden ejecutarse, aunque haya suficiente memoria disponible en el conjunto.

#### Particiones Dinámicas

- La memoria se divide en bloques de tamaño variable que se crean según la necesidad de los procesos.
- Los procesos se cargan en la memoria de manera continua, ocupando solo el espacio que necesitan.

#### **Ventajas**

- **Menor fragmentación interna**, ya que el tamaño de las particiones se adapta al tamaño del proceso.
- Mayor flexibilidad para acomodar procesos de distintos tamaños.

#### <u>Desventajas</u>

- **Fragmentación interna**: Espacios libres pequeños se acumulan entre particiones, dificultando la asignación de nuevos procesos.
- Mayor sobrecarga del sistema operativo, que debe realizar un seguimiento continuo de las áreas libres y ocupadas.

Característica	Particiones Fijas	Particiones Dinámicas
Tamaño de las particiones	Estático, definido al inicio	Variable, según las necesidades
Fragmentación	Interna	Externa
Flexibilidad	Limitada	Alta
Complejidad de gestión	Baja	Alta

#### b) Información requerida por el SO para administrar memoria

#### Particiones fijas

- Número de particiones y sus tamaños.
- Estado de cada partición (libre u ocupada).
- Dirección base de cada partición para la traducción de direcciones.

#### Particiones dinámicas

- Lista de espacios libres (dirección base y tamaño de cada bloque).
- Información sobre los procesos cargados (dirección base y tamaño).
- Mecanismos para manejar la fragmentación externa, como la compactación.

#### Partición fija

#### Partición dinámica

- 3) Al trabajar con particiones fijas, los tamaños de las mismas se pueden considerar:
  - Particiones de igual tamaño.
  - Particiones de diferente tamaño.

Cite ventajas y desventajas de estos 2 métodos.

i) Particiones de igual tamaño:

#### Ventajas:

- **Simplicidad en la administración**: Es más fácil de implementar porque el sistema no necesita calcular o manejar particiones de diferentes tamaños.
- **Baja sobrecarga**: El sistema operativo realiza menos operaciones para administrar estas particiones.
- **Asignación rápida**: Un proceso se asigna a la primera partición disponible, sin la necesidad de buscar un tamaño específico.

#### Desventajas:

- Fragmentación interna significativa: Si un proceso es mucho más pequeño que el tamaño de la partición, el espacio sobrante dentro de la partición no se puede usar.
- Ineficiencia para procesos grandes: Procesos que necesitan más memoria que el tamaño de una partición no pueden ejecutarse, aunque haya memoria disponible en otras particiones.

#### ii) Particiones de diferente tamaño:

#### Ventajas:

- **Menor fragmentación interna**: Se pueden asignar procesos a particiones cuyo tamaño sea más cercano al requerido, aprovechando mejor el espacio.
- Mejor adaptabilidad: Los procesos pequeños y grandes tienen mayor probabilidad de ser acomodados.

#### Desventajas:

- Complejidad en la administración: El sistema operativo debe buscar una partición del tamaño adecuado para cada proceso, aumentando el tiempo de asignación.
- Fragmentación externa: Si las particiones no se utilizan correctamente, pueden quedar muchas partes pequeñas de memoria libre que no sean útiles para los procesos.

Característica	Particiones de igual tamaño	Particiones de diferente tamaño
Fragmentación interna	Alta	Baja
Fragmentación externa	Baja	Moderada
Simplicidad de administración	Alta	Baja
Aprovechamiento de memoria	Вајо	Mejor
Asignación de procesos	Rápida, sin considerar tamaños	Más lenta debido a la búsqueda de partición adecuada

#### 4) Fragmentación:

Ambos métodos de particiones presentan el problema de la fragmentación:

- Fragmentación Interna (Para el caso de Particiones Fijas).
- Fragmentación Externa (Para el caso de Particiones Dinámicas).
- a) Explique a qué hacen referencia estos 2 problemas.
- **b)** El problema de la Fragmentación Externa es posible subsanar. Explique una técnica que evite este problema.

<u>Fragmentación Interna</u>: Ocurre en particiones fijas cuando un proceso no utiliza todo el espacio asignado en una partición.

**Causa**: Los procesos ocupan menos espacio que el tamaño de la partición, dejando memoria inutilizada dentro de la misma.

<u>Fragmentación Externa</u>: Ocurre en particiones dinámicas cuando la memoria disponible no es contigua, aunque en total sea suficiente para satisfacer la solicitud de un proceso.

**Causa**: Los procesos que se asignan y liberan crean pequeños huecos entre los bloques de memoria, lo que dificulta encontrar un espacio continuo para nuevos procesos.

El problema de la fragmentación externa se puede abordar con la técnica de **compactación de memoria**.

Consiste en reorganizar los procesos que están cargados en memoria para juntar los bloques libres en un solo espacio contiguo.

- i) El sistema operativo pausa momentáneamente la ejecución de los procesos.
- ii) Los datos de los procesos se mueven hacia un extremo de la memoria.
- iii) Se consolida todo el espacio libre en una única región contigua.

#### Ventajas:

- Liberar espacio contiguo para acomodar nuevos procesos, evitando problemas de fragmentación externa.
- Permite una mejor utilización de la memoria.

#### **Desventajas**:

- Es costosa en términos de tiempo y rendimiento, ya que requiere mover datos en memoria.
- No es adecuada para sistemas en tiempo real.

Alternativa moderna: Técnica de paginación.

En lugar de particiones dinámicas, los sistemas modernos suelen usar **paginación** o segmentación:

- Dividen la memoria en bloques más pequeños (páginas o segmentos), eliminando la necesidad de bloques contiguos.
- Con esta técnica, el problema de la fragmentación externa prácticamente desaparece, aunque puede persistir una mínima fragmentación interna.

#### 5) Paginación:

- a) Explique cómo trabaja este método de asignación de memoria.
- **b)** ¿Qué estructuras adicionales debe poseer el SO para llevar a cabo su implementación?
- c) Explique, utilizando gráficos, como son transformadas las direcciones lógicas en físicas
- d) En este esquema: ¿Se puede producir fragmentación (interna y/o externa)?

La **paginación** es un método de asignación de memoria que divide tanto la memoria física como la memoria lógica en bloques de tamaño fijo llamados marcos (frames) y páginas, respectivamente. Este esquema permite que los procesos no necesiten espacio contiguo en memoria física, mejorando el aprovechamiento.

#### Funcionamiento:

#### i) División del espacio:

- La memoria física se divide en marcos de tamaño fijo.
- La memoria lógica de cada proceso se divide en páginas del mismo tamaño que los marcos.

#### ii) Asignación:

- Las páginas del proceso se asignan a marcos libres en la memoria física.
- No es necesario que las páginas estén en marcos contiguos.

#### iii) Traducción de direcciones:

- Cada dirección lógica generada por el proceso consta de:
  - Un número de página (que identifica qué página lógica contiene los datos).
  - Un desplazamiento (offset), que indica la ubicación exacta dentro de esa página.
- El sistema operativo utiliza una tabla de páginas para traducir la dirección lógica en dirección física.

#### iv) Bit de presencia/ausencia:

 Indica si una página lógica está cargada en memoria física o debe traerse desde el disco.

#### Transformación de direcciones:

- Tamaño de página = 4 KB (4096 bytes).
- Dirección lógica: 0x1A32.
  - Número de página: 1A32 ÷ 4096 = 1 (división entera → página 1).
  - Offset: 1A32 mod 4096 = 0x32 (resto → desplazamiento dentro de la página).

# Si la tabla de páginas indica que:

Página lógica 1 → Marco físico 5.

## Entonces, la dirección física es:

Dirección física: Marco físico 5 × Tamaño de página + Offset = 5 × 4096 +
 0x32 = 0x5032.

```
      Proceso lógico:
      Memoria física:

      +----+
      +----+

      | Página 0|---->(Tabla)---->| Marco 3|

      +----+
      +-----+

      | Página 1|---->(Tabla)---->| Marco 5|

      +-----+
      | Página 2|---->(Tabla)---->| Marco 1|

      +-----+
      +-----+
```

#### Fragmentación en paginación:

#### Interna:

 Puede ocurrir si un proceso no utiliza completamente la última página asignada. El espacio restante dentro del marco queda desperdiciado.

#### Externa:

 No ocurre, ya que las páginas se asignan a marcos independientes de su ubicación en memoria física, eliminando la necesidad de bloques contiguos.

La paginación elimina la fragmentación externa, pero puede introducir una pequeña cantidad de fragmentación interna debido al tamaño fijo de las páginas.

6) Cite similitudes y diferencias entre la técnica de paginación y la de particiones fijas.

Similitudes entre Paginación y Particiones Fijas	
Aspecto	Descripción
División en bloques de tamaño fijo	Ambos dividen la memoria en unidades de tamaño fijo (particiones o marcos/páginas).
Rendimiento predecible	El acceso a memoria es rápido y constante, ya que no depende de la ubicación contigua.
Fragmentación interna	Puede generarse si el proceso no utiliza completamente el bloque asignado.
Adecuación para tamaños conocidos	Ambos funcionan bien cuando los tamaños de los procesos son uniformes o predecibles.

# Diferencias entre Paginación y Particiones Fijas

Aspecto	Paginación	Particiones Fijas
Tamaño de los bloques	Las páginas y marcos son de tamaño fijo, definido por el sistema.	Las particiones son de tamaño fijo, predefinido al configurar el sistema.
Flexibilidad	No requiere memoria contigua, las páginas se asignan a marcos disponibles.	Requiere que el proceso completo quepa en una partición fija.
Fragmentación externa	No ocurre debido a la asignación no contigua de páginas.	Puede ocurrir si no hay bloques contiguos suficientes aunque haya memoria libre.
Fragmentación interna	Puede ocurrir en la última página de un proceso si no está completamente llena.	Ocurre cuando un proceso no llena completamente su partición asignada.
Gestión de procesos	Admite procesos de cualquier tamaño, dividiéndolos en páginas.	Limita los procesos al tamaño máximo de una partición fija.
Estructuras adicionales	Requiere tabla de páginas y hardware adicional (MMU) para mapear direcciones.	No requiere estructuras adicionales complejas.
Complejidad de implementación	Más compleja debido a la traducción de direcciones y estructuras requeridas.	Más simple de implementar.
Tamaño ajustable	Adapta dinámicamente el uso de memoria según las necesidades del proceso.	Los tamaños de las particiones son fijos y no adaptables a procesos más grandes.

#### 9) Segmentación

- a) Explique cómo trabaja este método de asignación de memoria.
- **b)** ¿Qué estructuras adicionales debe poseer el SO para llevar a cabo su implementación?
- **c)** Explique, utilizando gráficos, como son transformadas las direcciones lógicas en lógicas.
- d) En este esquema: ¿Se puede producir fragmentación (interna y/o externa)?

La segmentación es un método de asignación de memoria que divide la memoria lógica de un proceso en bloques de tamaño variable llamados **segmentos**. Cada segmento corresponde a una unidad lógica de datos, como el código, datos, pila, etc.

#### Cómo trabaja la segmentación

#### i) División lógica:

- La memoria de cada proceso se organiza en segmentos según su naturaleza o función. Por ejemplo:
  - Segmento 0: Código del programa.
  - Segmento 1: Datos del programa.
  - Segmento 2: Pila del programa.
- Cada segmento puede tener un tamaño diferente, dependiendo de las necesidades del proceso.

# ii) Dirección lógica:

- Una dirección lógica consta de dos partes:
  - Número de segmento: Identifica el segmento al que pertenece la dirección
  - Desplazamiento (offset): Es la posición específica dentro del segmento.

#### iii) Traducción de direcciones:

- Cada segmento tiene una base (dirección inicial en memoria física) y un límite (tamaño del segmento).
- El sistema operativo verifica que el desplazamiento esté dentro del límite del segmento y calcula la dirección física sumando el desplazamiento a la base.

#### Estructuras adicionales necesarias en el SO

#### i) Tabla de segmentos:

- Cada proceso tiene una tabla que almacena:
  - Base: Dirección inicial de cada segmento en la memoria física.
  - Límite: Tamaño del segmento.

#### ii) Registro de base de la tabla de segmentos (STBR):

- Apunta al inicio de la tabla de segmentos del proceso actual.

- iii) Unidad de gestión de memoria (MMU):
  - Hardware que realiza la traducción de direcciones lógicas a físicas utilizando la tabla de segmentos.

# Ejemplo

#### Supongamos:

- Un proceso tiene tres segmentos:
  - Segmento 0 (Código): Base = 1000, Límite = 400.
  - Segmento 1 (Datos): Base = 2000, Límite = 500.
  - Segmento 2 (Pila): Base = 3000, Límite = 300.
- Dirección lógica: (Segmento 1, Offset 150).
  - Número de segmento: 1 (Datos).
  - Offset: 150.

#### Traducción:

- 1. Se verifica que el offset (150) está dentro del límite del segmento (500).
- 2. Dirección física = Base del segmento + Offset = 2000 + 150 = 2150.

**10)** Cite similitudes y diferencias entre la técnica de segmentación y la de particiones dinámicas.

Similitudes entre Segmentación y Particiones Dinámicas	
Aspecto	Descripción
Tamaños variables	Ambas dividen la memoria en bloques de tamaño variable, adaptándose a las necesidades de los procesos.
Asignación contigua	En ambos casos, cada bloque asignado (segmento o partición) ocupa un espacio contiguo en memoria física.
Fragmentación externa	Ambas técnicas pueden generar fragmentación externa debido a la asignación y liberación de bloques.
Compactación necesaria	En ambos casos, la compactación de memoria puede ser necesaria para reducir la fragmentación externa.
Flexibilidad	Permiten una mejor adaptación al tamaño real de los procesos, optimizando el uso de memoria en comparación con particiones fijas.

Diferencias entre Segmentación y Particiones Dinámicas		
Aspecto	Segmentación	Particiones Dinámicas
Unidad de asignación	Divide la memoria lógica del proceso en segmentos funcionales (código, datos, pila, etc.).	Divide la memoria física en bloques variables según el tamaño total del proceso.
Estructuras adicionales	Requiere una tabla de segmentos para mapear segmentos lógicos a direcciones físicas.	No requiere estructuras complejas, solo un índice para rastrear bloques libres y asignados.
Organización lógica	Basada en unidades lógicas del programa.	Basada en asignar el espacio necesario para todo el proceso.
Traducción de direcciones	Utiliza la <b>tabla de segmentos</b> para transformar direcciones lógicas a físicas.	No realiza traducción de direcciones, las direcciones son físicas directamente.
Fragmentación interna	No ocurre, ya que los segmentos se ajustan a su tamaño exacto.	Puede ocurrir si se asigna más memoria de la necesaria debido al diseño del bloque.
Complejidad	Más compleja, requiere gestión de segmentos y hardware adicional (MMU).	Más simple, solo se necesita gestionar los bloques dinámicos asignados.

11) Cite similitudes y diferencias entre la técnica de paginación y segmentación.

# Similitudes entre Paginación y Segmentación Aspecto Descripción División de la memoria Ambas dividen la memoria en bloques para gestionar procesos y asignar memoria. Traducción de En ambas técnicas, las direcciones lógicas deben ser direcciones traducidas a direcciones físicas mediante estructuras específicas (tablas). Flexibilidad en la No requieren que todo el proceso esté en memoria contigua, asignación lo que permite un mejor uso del espacio disponible. Eliminación de Ambas técnicas evitan la fragmentación externa, aunque la fragmentación externa segmentación puede causarla bajo ciertas condiciones. Uso de hardware Ambas requieren soporte de hardware, como la MMU adicional (Unidad de Gestión de Memoria), para la traducción de direcciones.

# Diferencias entre Paginación y Segmentación

Aspecto	Paginación	Segmentación
Unidad de asignación	Divide la memoria en bloques de tamaño fijo llamados páginas y marcos.	Divide la memoria en bloques de tamaño variable llamados segmentos.
Organización lógica	No sigue la estructura lógica del programa; se basa en divisiones de tamaño fijo.	Se basa en unidades funcionales del programa, como código, datos y pila.
Tamaño de los bloques	Las páginas y marcos tienen un tamaño fijo definido por el sistema.	Los segmentos tienen tamaños variables según las necesidades del proceso.
Fragmentación interna	Puede ocurrir en la última página si no está completamente utilizada.	No ocurre, ya que los segmentos se ajustan al tamaño exacto requerido.
Fragmentación externa	No ocurre, ya que las páginas pueden estar en cualquier marco disponible.	Puede ocurrir, ya que los segmentos requieren memoria contigua.
Estructuras necesarias	Requiere una <b>tabla de páginas</b> para mapear páginas lógicas a marcos físicos.	Requiere una <b>tabla de</b> <b>segmentos</b> con base y límite para cada segmento.
Complejidad de implementación	Más simple debido a los tamaños fijos de las páginas y marcos.	Más compleja por el manejo de segmentos de tamaños variables.
Traducción de direcciones	Utiliza el número de página y el desplazamiento dentro de la página.	Utiliza el número de segmento y el desplazamiento dentro del segmento.

#### 13) Memoria Virtual

- a) Describa qué beneficios introduce este esquema de administración de la memoria.
- b) ¿En qué se debe apoyar el SO para su implementación?
- c) Al implementar esta técnica utilizando paginación por demanda, las tablas de páginas de un proceso deben contar con información adicional además del marco donde se encuentra la página. ¿Cuál es esta información? ¿Por qué es necesaria?

La **memoria virtual** es una técnica que permite que los procesos aparenten tener más memoria disponible de la que físicamente existe en el sistema, utilizando almacenamiento secundario (como el disco) como extensión de la memoria RAM.

#### **Beneficios**

Beneficio	Descripción
Ejecución de procesos grandes	Permite ejecutar procesos que requieren más memoria de la que físicamente está disponible.
Multiprogramación	Facilita la ejecución concurrente de múltiples procesos, incluso si la memoria física es limitada.
Uso eficiente de la memoria	La memoria física se utiliza solo para las partes del programa que están activamente en uso.
Protección y aislamiento	Cada proceso tiene su espacio de memoria virtual, evitando interferencias entre procesos.
Simplificación del desarrollo	Los programadores no necesitan preocuparse por la ubicación física de los datos en memoria.
Flexibilidad en la asignación	Combina dinámicamente memoria física y almacenamiento secundario, ajustándose a las necesidades del sistema.

#### Elementos necesarias para implementar la memoria virtual:

#### i) Paginación o Segmentación:

 Técnicas como la paginación por demanda o la segmentación permiten implementar la memoria virtual, gestionando qué partes del programa están en memoria física y cuáles permanecen en el disco.

## ii) Unidad de Gestión de Memoria (MMU):

- Hardware que traduce direcciones virtuales a direcciones físicas mediante tablas (páginas o segmentos).

#### iii) Tabla de páginas:

 Para mapear las páginas lógicas del proceso a los marcos en memoria física o a posiciones en disco.

#### iv) Archivo de intercambio (SWAP):

- Espacio en el disco utilizado como extensión de la memoria RAM para almacenar páginas no activas.

#### v) Algoritmos de reemplazo de páginas:

- Para decidir qué páginas de memoria física deben ser intercambiadas al disco en caso de falta de espacio (LRU, FIFO, etc).

#### vi) Manejador de fallos de página:

- Parte del SO que gestiona las interrupciones cuando un proceso intenta acceder a una página que no está en memoria física.

#### Información adicional en las tablas de páginas:

Además de mapear las páginas a los marcos, las tablas de páginas de un proceso que utiliza paginación por demanda deben contener la siguiente información adicional:

Campo	Descripción
Presencia/ausencia	Indica si la página está en memoria física o en el disco (bit de presencia).
Dirección en disco	Si la página no está en memoria, señala su ubicación en el archivo de intercambio (swap).
Bits de acceso	Indican si la página ha sido leída o escrita recientemente (ayuda en los algoritmos de reemplazo).
Bit de modificación	Indica si la página ha sido modificada, lo que determina si necesita ser escrita de vuelta al disco.
Permisos de acceso	Define los permisos (lectura, escritura, ejecución) para proteger la memoria frente a accesos indebidos.
Tamaño de la página	Especifica el tamaño de la página (en sistemas que admiten páginas de tamaño variable).

#### Esta información es necesaria debido a:

#### i) Gestión de fallos de página:

- El bit de presencia permite al sistema identificar rápidamente si una página está en memoria física o necesita cargarse desde el disco.

#### ii) Protección de memoria:

- Los permisos de acceso garantizan que las operaciones en memoria respeten las restricciones del proceso.

#### iii) Optimización del reemplazo de páginas:

 Los bits de acceso y modificación ayudan a los algoritmos de reemplazo a tomar decisiones eficientes sobre qué páginas intercambiar al disco.

#### iv) Ubicación en disco:

- La dirección en disco permite localizar rápidamente una página cuando necesita ser cargada a memoria física.

#### 14) Fallos de Página (Page Faults):

- a) ¿Cuándo se producen?
- b) ¿Quién es responsable de detectar un fallo de página?
- c) Describa las acciones que emprende el SO cuando se produce un fallo de página.

Un **page fault** ocurre cuando un proceso intenta acceder a una página que no se encuentra en la memoria física (RAM) en ese momento y debe ser cargada desde el disco o archivo de intercambio (SWAP).

Los page fault se producen cuando:

#### Una página no está en memoria física:

 El proceso accede a una dirección virtual cuya página no tiene un marco asignado en RAM.

#### Primera referencia a una página:

 Cuando un proceso accede por primera vez a una página aún no cargada en memoria.

#### Paginación por demanda:

 Cuando se implementa memoria virtual y solo se cargan las páginas requeridas en el momento de uso, dejando otras en el disco hasta que sean necesarias.

El responsable de detectar un page fault es el hardware, específicamente la MMU (Unidad de Gestión de Memoria), en colaboración con el SO:

#### MMU:

- Al traducir una dirección virtual a física, verifica la tabla de páginas.
- Si el bit de presencia de la página está desactivado (indicando que no está en memoria física), genera una interrupción de fallo de página.

#### SO:

 Recibe la interrupción de fallo de página y ejecuta las acciones necesarias para resolverlo.

Cuando se produce un page fault, el SO sigue estos pasos:

#### i) Interrupción del proceso:

Suspende el proceso que causó el fallo de página.

#### ii) Verificación de la dirección:

- Comprueba si la dirección solicitada es válida:
  - Si no es válida, genera un error de segmentación o acceso no permitido.
  - Si es válida, continúa con los siguientes pasos.

#### iii) Ubicación de la página en disco:

 Consulta la tabla de páginas para localizar la posición de la página en el archivo de intercambio (SWAP) o disco.

# iv) Liberación de un marco (si es necesario):

- Si no hay marcos libres en memoria física, selecciona uno para liberar usando un algoritmo de reemplazo de páginas (como FIFO, LRU, etc).

#### v) Escritura de página modificada:

- Si la página seleccionada para liberar ha sido modificada (bit de modificación activo), se escribe de vuelta al disco para preservar los cambios.

#### vi) Carga de la página en memoria física:

- Se carga la página solicitada desde el disco al marco liberado en memoria física.

#### vii) Actualización de la tabla de páginas:

- Actualiza la entrada correspondiente en la tabla de páginas:
  - Activa el bit de presencia.
  - Indica el marco asignado en memoria física.

#### viii) Reanudación del proceso:

 Reactiva el proceso interrumpido, que puede continuar su ejecución sin saber que ocurrió el fallo.

Paso	Descripción
1. Interrupción del proceso	El sistema operativo suspende el proceso que causó el fallo de página.
2. Verificación de la dirección	Se evalúa si la dirección solicitada es válida: - Inválida: genera un error o finaliza el proceso. - Válida: continúa con el flujo.
3. Ubicación de la página	Se busca la posición de la página en el archivo de intercambio (swap) o en el disco.
4. Liberación de un marco	Si no hay marcos disponibles:  - Se selecciona un marco con un <b>algoritmo de reemplazo de páginas</b> (FIFO, LRU, etc.).  - Si el marco contiene una página modificada, se guarda en disco antes de liberarlo.
5. Carga de la página	Se carga la página requerida desde el disco a un marco disponible en memoria física.
6. Actualización y reanudación	La tabla de páginas del proceso se actualiza: - Se activa el <b>bit de presencia</b> . - Se registra el marco asignado. Finalmente, el proceso se reactiva y continúa su ejecución.

# 15) Direcciones:

- **a)** Si se dispone de un espacio de direcciones virtuales de 32 bits, donde cada dirección referencia 1 byte:
  - i) ¿Cuál es el tamaño máximo de un proceso (recordar "espacio virtual")?
  - ii) Si el tamaño de página es de 512kb. ¿Cuál es el número máximo de páginas que puede tener un proceso?
  - iii) Si el tamaño de página es de 512kb, y se disponen de 256mb de memoria real. ¿Cuál es el número de marcos que puede haber?
  - iv) Si se utilizaran 2kb para cada entrada en la tabla de páginas de un proceso: ¿Cuál sería el tamaño máximo de la tabla de páginas de cada proceso?

# a) Espacio de direcciones virtuales de 32 bits

# i) ¿Cuál es el tamaño máximo de un proceso?

El espacio de direcciones virtuales es de 32 bits, lo que significa que el sistema puede direccionar hasta  $2^{32}$  bytes.

Cálculo:

Tamaño máximo de un proceso  $= 2^{32}$  bytes = 4,294,967,296 bytes = 4 GB

Respuesta: El tamaño máximo de un proceso es 4 GB.

# ii) ¿Cuál es el número máximo de páginas que puede tener un proceso si el tamaño de página es

El tamaño de la página es de 512 KB (kilobytes), lo cual equivale a 524,288 bytes.

Cálculo

$$\text{N\'umero de p\'aginas} = \frac{\text{Tama\~no m\'aximo del espacio virtual}}{\text{Tama\~no de la p\'agina}} = \frac{2^{32} \text{ bytes}}{512 \times 1024 \text{ bytes}} = \frac{4,294,967,296}{524,288} = 8192$$

Respuesta: El número máximo de páginas que puede tener un proceso es 8192 páginas.

# iii) Si el tamaño de página es de 512 KB y se disponen de 256 MB de memoria real, ¿cuál es el número de marcos que puede haber?

El tamaño de la página es de **512 KB** (kilobytes), y la memoria real disponible es de **256 MB** (megabytes), que equivale a **262,144 KB** o **268,435,456 bytes**.

Cálculo:

$$\label{eq:numero} \text{N\'umero de marcos} = \frac{\text{Memoria real disponible}}{\text{Tama\~no de la p\'agina}} = \frac{256 \times 1024 \text{ KB}}{512 \text{ KB}} = \frac{262,144 \text{ KB}}{512 \text{ KB}} = 512$$

Respuesta: El número de marcos que puede haber en la memoria real es 512 marcos.

#### iv) ¿Cuál sería el tamaño máximo de la tabla de páginas si se utilizaran 2 KB para cada entrada?

Cada entrada de la tabla de páginas ocupa 2 KB (kilobytes), y sabemos que el número máximo de páginas es 8192 (calculado previamente).

• Cálculo:

 $Tamaño \ de \ la \ tabla \ de \ páginas = N\'umero \ de \ p\'aginas \times Tamaño \ de \ cada \ entrada = 8192 \times 2 \ KB = 16,384 \ KB = 16 \ MB$ 

Respuesta: El tamaño máximo de la tabla de páginas sería 16 MB.

Pregunta	Fórmula	Descripción
i) Tamaño máximo de un proceso	Tamaño máximo de un proceso $=2^{ m Nbits}$	Para un espacio de direcciones virtuales de $N$ bits, el tamaño máximo es $2^N$ bytes.
ii) Número máximo de páginas	Número de páginas = <u>Tamaño máximo del espacio virtual</u> Tamaño de la página	Se divide el espacio virtual total por el tamaño de la página.
iii) Número de marcos en memoria real	$ ext{Número de marcos} = rac{ ext{Memoria real disponible}}{ ext{Tamaño de la página}}$	Se divide la memoria física disponible por el tamaño de la página.
iv) Tamaño de la tabla de páginas	Tamaño de la tabla de páginas = Número de páginas × Tamaño de cada entrada	Se multiplica el número de páginas por el tamaño de cada entrada en la tabla.

**16)** Como se vio en el ejercicio anterior, la tabla de páginas de un proceso puede alcanzar un tamaño considerablemente grande, que incluso, no podría almacenarse de manera completa en la memoria real. Es por esto que el SO también realiza paginación sobre las tablas de páginas.

Existen varios enfoques para administrar las tablas de páginas:

- Tablas de páginas de 1 nivel.
- Tablas de páginas de 2 niveles.
- Tablas de páginas invertidas.

Explique brevemente cómo trabajan estos enfoques e indique cómo se realiza la transformación de la dirección virtual en dirección física.

#### Tablas de páginas de 1 nivel:

En este enfoque, la **tabla de páginas** es una tabla que almacena la dirección física de cada página en el espacio de direcciones virtuales.

#### Estructura:

- La dirección virtual se divide en dos partes:
  - Índice de página: Indica cuál entrada de la tabla de páginas se debe consultar.
  - **Desplazamiento dentro de la página**: Indica la ubicación dentro de la página en la memoria física.
- La tabla de páginas tiene una entrada por cada página en el espacio de direcciones virtuales.

#### Transformación de la dirección virtual a física:

- Se toma el índice de página y se usa para buscar la dirección física de la página en la tabla de páginas.
- Se combina esta dirección con el desplazamiento (la parte de la dirección virtual que indica la posición dentro de la página) para obtener la dirección física final.

**Problema**: Si el número de páginas es muy grande, la tabla de páginas puede ser muy grande para almacenarse completamente en la memoria real.

#### Tablas de páginas de 2 niveles:

En este enfoque, se usan dos niveles de tablas de páginas, lo que ayuda a reducir el tamaño de la tabla de páginas al hacerla más jerárquica.

#### Estructura:

- Se divide la dirección virtual en tres partes:
  - Índice de nivel superior: Apunta a una tabla de páginas de nivel superior (tabla de primer nivel).
  - **Índice de nivel inferior**: Apunta a la tabla de páginas de nivel inferior (tabla de segundo nivel).
  - Desplazamiento dentro de la página: Indica la ubicación dentro de la página en la memoria física.
- La tabla de páginas de nivel superior contiene direcciones físicas de las tablas de segundo nivel, y cada tabla de segundo nivel contiene las direcciones físicas de las páginas de memoria.

#### Transformación de la dirección virtual a física:

- i) Se utiliza el **índice de nivel superior** para acceder a una entrada en la tabla de primer nivel.
- **ii)** Luego, el índice de nivel inferior se usa para acceder a la tabla de segundo nivel.
- **iii)** Finalmente, el desplazamiento se suma a la dirección física obtenida para encontrar la dirección final en memoria.

**Ventaja**: Reduce el tamaño de la tabla de páginas principal, ya que no es necesario mantener todas las direcciones de las páginas en memoria, sino solo las direcciones de las tablas de segundo nivel que realmente están en uso.

#### Tablas de páginas invertidas:

En este enfoque, se utiliza una tabla de páginas invertida que es diferente de las tablas tradicionales de páginas.

#### Estructura:

- La tabla de páginas invertida tiene una sola entrada por página física (en lugar de por página virtual).
- Cada entrada almacena la dirección del **proceso** y la **página virtual** correspondiente a una página física específica.

#### Transformación de la dirección virtual a física:

- i) La dirección virtual se divide en dos partes:
  - Índice de página virtual: Esto se usa para buscar en la tabla de páginas invertida.
  - **Desplazamiento**: El desplazamiento dentro de la página no cambia, se mantiene igual en la dirección física.
- ii) La tabla de páginas invertida se consulta para encontrar qué proceso tiene asignada esa página virtual a una página física.
- iii) El desplazamiento se suma a la página física encontrada para obtener la dirección final.

**Ventaja**: La tabla de páginas invertida reduce el tamaño de la tabla de páginas al tener solo una entrada por cada página física, independientemente de cuántos procesos estén usando la memoria.

Enfoque	Pasos para la Transformación
1. Tablas de Páginas de 1	<ol> <li>Usar el índice de página para acceder a la entrada de la</li></ol>
Nivel	tabla de páginas. <li>Sumar el desplazamiento a la dirección física.</li>
2. Tablas de Páginas de 2 Niveles	<ol> <li>Usar el índice de nivel superior para acceder a la tabla de primer nivel.</li> <li>Usar el índice de nivel inferior para acceder a la tabla de segundo nivel.</li> <li>Sumar el desplazamiento.</li> </ol>
3. Tablas de Páginas	<ol> <li>Usar el índice de la página virtual para buscar en la tabla</li></ol>
Invertidas	de páginas invertida. <li>Sumar el desplazamiento a la dirección física obtenida.</li>

# 18) Tamaño de la página:

La selección del tamaño de la página influye de manera directa sobre el funcionamiento de la memoria virtual. Compare las siguientes situaciones con respecto al tamaño de página, indicando ventajas y desventajas:

- Un tamaño de página pequeño.
- Un tamaño de página grande.

Aspecto	Ventajas Desventajas	
Fragmentación interna	Menor fragmentación interna: Se desperdicia menos memoria dentro de las páginas.	
Precisión en la asignación	Mayor precisión: Permite asignar memoria de manera más precisa a procesos pequeños.	
Tablas de páginas		Tablas más grandes: Incrementa el tamaño de la tabla, ocupando más memoria.
Fallos de página		Mayor número de fallos: Más accesos al disco debido a mayor cantidad de páginas.
Gestión del SO		Mayor tiempo de gestión: Incrementa la carga del sistema operativo al manejar muchas páginas.

Aspecto	Ventajas	Desventajas
Fragmentación interna		Mayor fragmentación interna: Desperdicio de memoria para datos pequeños.
Precisión en la asignación		Menor precisión: Ineficiente para asignar páginas grandes a procesos pequeños.
Tablas de páginas	Más compactas: Menos páginas reducen el tamaño de la tabla, ahorrando memoria.	
Fallos de página	Menor número de fallos: Más datos por página reducen accesos frecuentes al disco.	
Gestión del SO	Eficiencia en el uso de disco: Acceso más eficiente en sistemas con alto rendimiento de I/O.	Uso ineficiente de memoria: Las páginas grandes pueden desperdiciar recursos en sistemas con muchos procesos pequeños.

## **19) Asignación de marcos a un proceso** (Conjunto de trabajo o Working Set):

Con la memoria virtual paginada, no se requiere que todas las páginas de un proceso se encuentren en memoria. El SO debe controlar cuantas páginas de un proceso puede tener en la memoria principal. Existen 2 políticas que se pueden utilizar:

- Asignación Fija
- Asignación DInámica
- a) Describa cómo trabajan estas 2 políticas.
- b) Dada la siguiente tabla de procesos y las páginas que ellos ocupan, y teniéndose 40 marcos en la memoria principal, cuántos marcos les corresponderían a cada proceso si se usa la técnica de Asignación Fija:
  - i) Reparto equitativo
  - ii) Reparto Proporcional

Proceso	Total de Paginas Usadas
1	15
2	20
3	20
4	8

c) ¿Cuál de los 2 repartos usados en b) resultó más eficiente? ¿Por qué?

Política	Descripción
Asignación Fija	<ul> <li>Cada proceso recibe una cantidad fija de marcos en memoria principal.</li> <li>Esta cantidad no cambia durante la ejecución del proceso.</li> <li>Puede basarse en un reparto equitativo o proporcional.</li> </ul>
Asignación Dinámica	<ul> <li>Los marcos asignados a un proceso pueden variar durante su ejecución.</li> <li>El SO ajusta dinámicamente el número de marcos asignados a cada proceso en función de su necesidad actual.</li> <li>Se basa en el Conjunto de Trabajo (Working Set), que identifica las páginas más utilizadas por el proceso en un período reciente.</li> </ul>

# i) Reparto Equitativo

• Fórmula:

$${\rm Marcos\ asignados\ por\ proceso} = \frac{{\rm Total\ de\ marcos\ disponibles}}{{\rm N\'umero\ de\ procesos}}$$

Cálculo:

$${\rm Marcos\ por\ proceso} = \frac{40}{4} = 10$$

Proceso	Marcos asignados (equidad)
1	10
2	10
3	10
4	10

#### ii) Reparto Proporcional

• Fórmula:

$${\rm Marcos\; para\; un\; proceso} = \left(\frac{{\rm P\'aginas\; del\; proceso}}{{\rm Total\; de\; p\'aginas\; de\; todos\; los\; procesos}}\right) \times {\rm Marcos\; disponibles}$$

Cálculo del total de páginas usadas por todos los procesos:

Total de páginas = 
$$15 + 20 + 20 + 8 = 63$$

Marcos para cada proceso:

Proceso	Cálculo	Marcos asignados (proporcional)
1	$\left(rac{15}{63} ight) imes40pprox9.52$	10
2	$\left(rac{20}{63} ight) imes40pprox12.70$	13
3	$\left(rac{20}{63} ight) imes40pprox12.70$	13
4	$\left(rac{8}{63} ight) imes40pprox5.08$	4

Aspecto	Reparto Equitativo	Reparto Proporcional	
Uso eficiente de memoria	No considera el tamaño real de los procesos.	Asigna más marcos a procesos con mayores necesidades.	
Fragmentación	Podría generar desperdicio en procesos pequeños.	Minimiza la fragmentación al considerar las necesidades reales.	
Carga de procesos grandes	Procesos grandes (como 2 y 3) podrían experimentar más fallos de página.	Procesos grandes tienen más marcos, reduciendo fallos de página.	

#### 20) Reemplazo de páginas (selección de una víctima):

¿Qué sucede cuando todos los marcos en la memoria principal están usados por las páginas de los procesos y se produce un fallo de página?

El SO debe seleccionar una de las páginas que se encuentra en memoria como víctima, y ser reemplazada por la nueva página que produjo el fallo.

Considere los siguientes algoritmos de selección de víctimas básicos:

- LRU
- FIFO
- OPT (Óptimo)
- Segunda Chance
- a) Clasifique estos algoritmos de malo a bueno de acuerdo a la tasa de fallos de página que se obtienen al utilizarlos.
- b) Analice su funcionamiento. ¿Cómo los implementaría?
- c) Sabemos que la página a ser reemplazada puede estar modificada. ¿Qué secciones debe llevar el SO cuando se encuentra ante esta situación?

Los algoritmos se clasifican según su eficiencia en la tasa de fallos de página, es decir, cuán bien gestionan las páginas en memoria para minimizar las veces que el sistema necesita cargar una nueva página desde el disco.

Algoritmo	Clasificación	Explicación
FIFO	Peor	First-In, First-Out (FIFO) es simple, pero a menudo no minimiza los fallos, ya que reemplaza las páginas sin considerar si son aún necesarias.
Segunda Chance	Moderado	Similar a FIFO pero con una mejora, da una segunda oportunidad a las páginas antes de reemplazarlas, lo que mejora la tasa de fallos en comparación con FIFO.
LRU	Bueno	Least Recently Used (LRU) reemplaza la página menos usada recientemente. Generalmente ofrece una buena tasa de aciertos en comparación con FIFO y Segunda Chance.
OPT	Мејог	Óptimo (OPT) reemplaza la página que no se necesitará por más tiempo en el futuro. Ofrece la mejor tasa de fallos, pero no es implementable en la práctica ya que requiere conocimiento del futuro.

#### Análisis de funcionamiento y cómo implementarlos:

#### **FIFO** (First-In, First-Out):

<u>Funcionamiento</u>: Las páginas se cargan en memoria y se almacenan en una cola. La primera página que entra es la primera en ser reemplazada cuando ocurre un fallo.

#### Implementación:

- Utilizar una queue que almacene las páginas en el orden en que entraron en memoria.
- Al producirse un fallo de página, simplemente se reemplaza la página en la cabeza de la queue (la más antigua).

#### Segunda Chance:

<u>Funcionamiento</u>: Es una mejora sobre FIFO. Cada vez que se produce un fallo de página, el sistema revisa si la página más antigua (en la cabeza de la queue) ha sido referenciada recientemente (a través de un bit de referencia).

- Si no ha sido referenciada, se reemplaza.
- Si ha sido referenciada, se le da una "segunda oportunidad" y mueve al final de la queue.

#### Implementación:

- Mantener un bit de referencia para cada página.
- Si el bit de referencia está activado (la página ha sido utilizada recientemente), se mueve al final de la queue.
- Si está desactivado, la página es reemplazada.

#### LRU (Least Recently Used):

<u>Funcionamiento</u>: Este algoritmo reemplaza la página que ha estado más tiempo sin ser utilizada.

#### Implementación:

- Se puede usar una lista vinculada o un contador de tiempo.
- Cada vez que se accede a una página, se actualiza su posición en la lista (la más reciente se mueve al principio).
- Cuando se produce un fallo de página, se reemplaza la página al final de la lista (la menos reciente).

#### **OPT** (Óptimo):

<u>Funcionamiento</u>: Este algoritmo reemplaza la página que no se necesitará por el mayor tiempo en el futuro. Dado que requiere conocer el futuro, es más un concepto teórico que una opción práctica.

## Implementación:

- Se necesitaría un conocimiento completo de las futuras referencias a páginas, lo cual no es posible en la práctica.
- Sin embargo, es útil como referencia para comparar otros algoritmos.

#### Acciones cuando la página a reemplazar está modificada:

Cuando el sistema va a reemplazar una página que ha sido modificada (también conocida como una página "sucia"), el sistema operativo debe asegurarse de que los cambios realizados en la página se persistan. Esto implica lo siguiente:

- i) Escribir la página en el disco: Si la página fue modificada en memoria (es decir, el bit de "modificado" o "sucio" está activado), debe guardarse en el disco (SWAP) antes de ser reemplazada. Esto asegura que los cambios no se pierdan.
- ii) Actualizar la tabla de páginas: Después de escribir la página modificada en el disco, el sistema operativo debe actualizar las tablas de páginas para reflejar que la página ha sido reemplazada.
- iii) Liberar el marco de memoria: Finalmente, el marco de memoria que contenía la página modificada debe ser liberado para poder asignarlo a una nueva página.

Este proceso se llama **escritura a disco** o **swapping** y puede causar una sobrecarga significativa en el rendimiento del sistema si se produce con frecuencia.

#### 21) Alcance del reemplazo:

Al momento de tener que seleccionar una página víctima, el SO puede optar por 2 política a utilizar.

- Reemplazo local
- Reemplazo global
- a) Describa cómo trabajan estas 2 políticas.
- b) ¿Es posible utilizar la política de "Asignación Fija" de marcos junto con la política de "Reemplazo Global"? Justifique.

#### Reemplazo local:

<u>Funcionamiento</u>: En el reemplazo local, cuando ocurre un fallo de página, el sistema operativo solo considera las páginas de un proceso específico para reemplazar una página. Es decir, cada proceso tiene su propio conjunto de marcos en memoria, y el reemplazo de páginas solo ocurre dentro de los marcos asignados a ese proceso.

<u>Ventaja</u>: Se garantiza que el proceso siempre tiene acceso a un conjunto de páginas fijo, lo que proporciona cierta estabilidad y previsibilidad.

<u>Desventaja</u>: Si un proceso necesita más memoria (por ejemplo, si tiene más páginas que las asignadas), puede experimentar muchos fallos de página, ya que no puede utilizar marcos de otros procesos.

#### Reemplazo global:

<u>Funcionamiento</u>: En el reemplazo global, el sistema operativo puede elegir cualquier página de la memoria (sin importar a qué proceso pertenezca) para ser reemplazada cuando ocurre un fallo de página. Esto significa que la selección de la página víctima puede incur páginas de cualquier proceso, no solo del proceso que causó el fallo.

<u>Ventaja</u>: Es más flexible, ya que permite reubicar páginas de cualquier proceso en la memoria, lo que puede reducir la tasa de fallos de páginas para los procesos más activos.

<u>Desventaja</u>: Puede generar problemas de "interferencia entre procesos", donde un proceso puede ser desfavorecido y sus páginas pueden ser reemplazadas con frecuencia para dar espacio a otros procesos.

No es posible utilizar **asignación fija de marcos** junto con **reemplazo global** de manera efectiva.

**Asignación fija de marcos** implica que cada proceso tiene una cantidad de marcos de memoria predeterminada, que no se puede modificar. Una vez que se asignan los marcos a un proceso, estos marcos no pueden ser utilizados por otros procesos, lo que significa que los procesos no pueden compartir espacio en memoria.

**Reemplazo global** implica que cualquier página de cualquier proceso puede ser reemplazada, lo que requiere la capacidad de reubicar páginas entre los marcos de todos los procesos. Si cada proceso tiene un conjunto fijo de marcos, no se puede aplicar el reemplazo global, ya que no hay marcos disponibles del conjunto asignado a un proceso.

Por lo tanto, **asignación fija** limita la flexibilidad necesaria para implementar **reemplazo global**, ya que el sistema no puede elegir páginas de otros procesos para ser reemplazadas. En este caso, el **reemplazo local** es la opción más adecuada, porque el reemplazo solo ocurre dentro de los marcos asignados a cada proceso.