1.- Explique a que hacen referencia los siguientes términos:

* Dirección Lógica o Virtual
* Dirección Física

Dirección lógica:

Es una dirección que enmascara o abstrae una dirección física (se usa para detallar cuanta memoria requiere el proceso total y así traducirla a memoria física con la MMU)

Referencia a una localidad en memoria

Se la debe traducir a una dirección física

Dirección Física:

Es la dirección real. Es con la que se accede efectivamente a memoria

Representa la dirección absoluta en memoria principal

Literalmente el lugar de la memoria RAM donde se aloja el proceso

2.- En la técnica de Particiones Múltiples, la memoria es divida en varias particiones y los procesos son ubicados en estas, siempre que el tamaño del mismo sea menor o igual que el tamaño de la partición. Al trabajar con particiones se pueden considerar 2 métodos (independientes entre si):

Particiones Fijas

Particiones Dinámicas

a) Explique cómo trabajan estos 2 métodos. Cite diferencias, ventajas y desventajas.

Particiones Fijas:

La memoria se divide en particiones o regiones de tamaño fijo → tamaños iguales o diferentes

Alojan un único proceso

Cada proceso se coloca en alguna partición de acuerdo a algún criterio:

• First Fit

• Best Fit

• Worst Fit

• Next Fit

Particiones Dinámicas:

Las particiones varían en tamaño y número

Alojan un proceso cada una

Cada partición se genera en forma dinámica del tamaño justo que necesita el proceso

Fijas la memoria en particiones en un tamaño determinado por el criterio, con esto podemos no tener fragmentación(interna), pero podemos dar mucho espacio/ poco espacio, cuando no es requerido. En cambio, con Dinámicas, se da un tamaño especifico que se necesita usar, pero puede generar fragmentación (externa)

b) ¿Qué información debe disponer el SO para poder administrar la memoria con estos métodos?

Para administrar la memoria con estos métodos, el sistema operativo debe tener información sobre la distribución de particiones en memoria, el estado de cada partición (ocupada o libre), el tamaño de las particiones, y qué proceso está en cada partición. Además, para particiones dinámicas, se requiere información sobre las direcciones base y límite de cada partición dinámica.

c) Realice un gráfico indicado como realiza el SO la transformación de direcciones lógicas a direcciones físicas.

la transformación de direcciones lógicas a direcciones físicas se realiza teniendo en cuenta tanto la dirección lógica del proceso como los límites o el tamaño de las particiones o segmentos de memoria en los que reside ese proceso. Esta transformación se lleva a cabo a través de la Unidad de Gestión de Memoria (MMU, por sus siglas en inglés).

La MMU es una parte del hardware que se encarga de realizar la traducción de direcciones lógicas a direcciones físicas. Para hacerlo, la MMU utiliza los registros de base y límite (base and limit registers) asociados a cada partición o segmento de memoria en el que se encuentra el proceso.

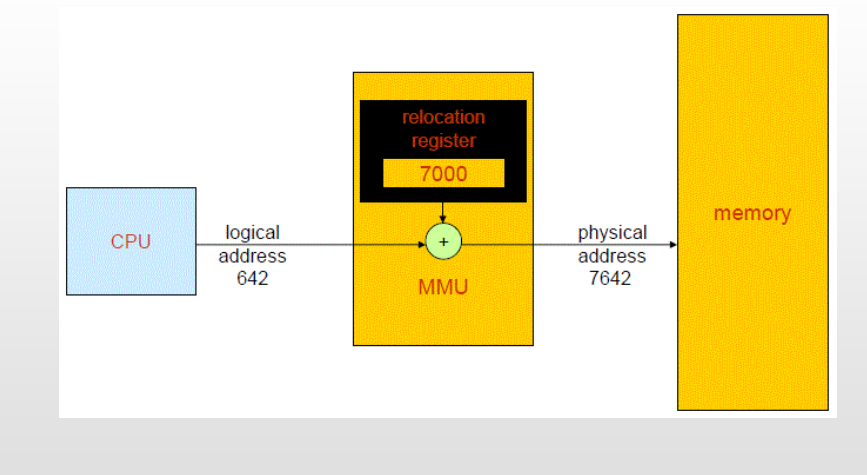
El proceso de traducción típico es el siguiente:

Cuando un proceso hace una referencia a memoria usando una dirección lógica, la MMU obtiene esta dirección lógica.

La MMU compara la dirección lógica con el límite de la partición o segmento del proceso en cuestión. Si la dirección lógica está dentro de los límites permitidos, se considera válida.

Luego, la MMU suma la dirección lógica con el valor almacenado en el registro de base de esa partición o segmento para obtener la dirección física correspondiente.

Este proceso de traducción garantiza que un proceso no pueda acceder a la memoria fuera de sus límites asignados, proporcionando protección y aislamiento entre los procesos. La MMU juega un papel fundamental en la administración de memoria y garantiza que los procesos no accedan a áreas de memoria que no les corresponden.



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

Particiones de Igual Tamaño (Fijas):

Ventajas:

Simplicidad: Es más fácil de implementar y administrar, ya que todas las particiones son del mismo tamaño.

Ausencia de Fragmentación Interna: Al ser todas las particiones del mismo tamaño, no hay fragmentación interna, lo que simplifica la gestión de la memoria.

Desventajas:

Desperdicio de Espacio: Puede haber un desperdicio significativo de memoria si los procesos son más pequeños que el tamaño de la partición, lo que lleva a una fragmentación externa.

Limitación en la Variedad de Procesos: Limita la capacidad del sistema para acomodar procesos de diferentes tamaños, lo que podría resultar en ineficiencia.

Particiones de Diferente Tamaño (Fijas):

Ventajas:

Uso Eficiente de la Memoria: Permite utilizar la memoria de manera más eficiente, ya que las particiones se ajustan al tamaño de los procesos, minimizando el desperdicio de espacio.

Mayor Flexibilidad: Puede acomodar una variedad de tamaños de procesos, lo que es beneficioso en entornos donde los procesos son de diferentes tamaños.

Desventajas:

Fragmentación Interna: Puede dar lugar a fragmentación interna, ya que las particiones de diferentes tamaños pueden dejar huecos no utilizados dentro de cada partición.

Complejidad de Gestión: La administración de particiones de diferentes tamaños es más compleja que las particiones de igual tamaño, ya que se requiere un seguimiento detallado de las particiones y su ocupación.

En resumen, las ventajas y desventajas de particiones de igual tamaño y particiones de diferente tamaño en el contexto de particiones fijas son las mismas que mencioné anteriormente. La elección entre estos dos métodos depende de las necesidades y restricciones específicas del sistema.

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 que hacen referencia estos 2 problemas

Fragmentación Interna:

La fragmentación interna ocurre cuando un bloque de memoria asignado a un proceso es más grande que el tamaño real del proceso. En otras palabras, es el espacio no utilizado dentro de un bloque de memoria asignado. Esto se debe a que los bloques de memoria asignados suelen tener un tamaño fijo, y si un proceso no ocupa todo ese espacio, se produce un desperdicio.

La fragmentación interna se asocia comúnmente con esquemas de asignación de memoria fija o particionada, donde se asignan bloques de memoria de un tamaño predefinido a los procesos. Si un proceso es más pequeño que la partición asignada, se produce fragmentación interna.

La fragmentación interna puede llevar a una utilización ineficiente de la memoria, ya que el espacio desperdiciado no puede utilizarse para otros fines.

Fragmentación Externa:

La fragmentación externa ocurre cuando hay suficiente memoria total disponible en el sistema, pero no se puede asignar un bloque contiguo lo suficientemente grande para satisfacer las necesidades de un proceso en particular. Esto se debe a que el espacio de memoria libre está fragmentado en múltiples bloques pequeños y dispersos.

La fragmentación externa es más común en sistemas con asignación de memoria dinámica, como en sistemas de particiones dinámicas, donde los bloques de memoria se crean y liberan dinámicamente. A medida que los procesos se asignan y liberan, se crean huecos en la memoria, lo que puede dificultar la asignación de bloques contiguos a procesos posteriores.

La fragmentación externa puede llevar a que no se puedan ejecutar procesos, a pesar de que haya suficiente memoria total disponible. Puede requerir estrategias de administración de memoria, como la compactación o la paginación, para mitigar este problema.

En resumen, la fragmentación interna se refiere al espacio no utilizado dentro de un bloque de memoria asignado a un proceso, mientras que la fragmentación externa se refiere a la dispersión de bloques de memoria libres que dificulta la asignación de bloques contiguos a procesos que los necesitan. Ambas formas de fragmentación pueden afectar negativamente la eficiencia y la utilización de la memoria en un sistema informático.

b) El problema de la Fragmentación Externa es posible de subsanar. Explique una técnica que evite este problema.

El proceso de compactación se puede describir de la siguiente manera:

Identificación de Fragmentación Externa: El sistema operativo monitorea continuamente el espacio de memoria y mantiene una lista de bloques de memoria asignados y bloques de memoria libres. Cuando detecta fragmentación externa, sabe que hay bloques de memoria libres dispersos que podrían usarse para alojar procesos si estuvieran contiguos.

Desplazamiento de Procesos: Para eliminar la fragmentación externa, el sistema operativo puede mover procesos de manera que los bloques de memoria ocupados estén adyacentes entre sí, creando un bloque de memoria contiguo más grande. Para hacer esto, el sistema operativo debe ajustar las direcciones lógicas de los procesos y actualizar las tablas de asignación de memoria.

Actualización de Registros de Base: Si se utilizan registros de base y límite en la administración de la memoria, se actualizarán los registros de base para reflejar la nueva ubicación de los procesos.

Liberación de Espacio No Utilizado: Una vez que los procesos se han reorganizado para eliminar la fragmentación externa, los bloques de memoria no utilizados que quedaron libres se pueden fusionar en un solo bloque de memoria libre más grande.

La compactación es una técnica efectiva para recuperar espacio fragmentado y garantizar que los procesos puedan asignarse bloques de memoria contiguos cuando sea necesario. Sin embargo, es importante destacar que la compactación puede ser un proceso costoso en términos de tiempo y recursos, ya que requiere mover procesos y actualizar estructuras de datos. Por lo tanto, generalmente se realiza en sistemas donde la fragmentación externa es un problema significativo y es menos invasivo en sistemas con fragmentación externa ocasional.

5.- Paginación

a) Explique cómo trabaja este método de asignación de memoria.

• La memoria se divide en porciones de igual tamaño llamadas marco

• El espacio de direcciones de los procesos se divide en porciones de igual tamaño denominadas páginas

• Tamaño pagina = tamaño marco = 512 bytes (generalmente)

• El SO mantiene una tabla de paginas para cada proceso, la cual contiene el marco donde se encuentra cada página

• La paginación bajo demanda es una técnica eficiente de manejar esta estrategia → Thrashing

b) ¿Qué estructuras adicionales debe poseer el SO para llevar a cabo su implementación?

La tabla de paginas.

c) Explique, utilizando gráficos, como son transformadas las direcciones lógicas en físicas.

• Un proceso en ejecución hace referencia a una dirección virtual → v = (p,d)

• El SO busca la página p en la tabla de páginas del proceso y determina en que marco se encuentra

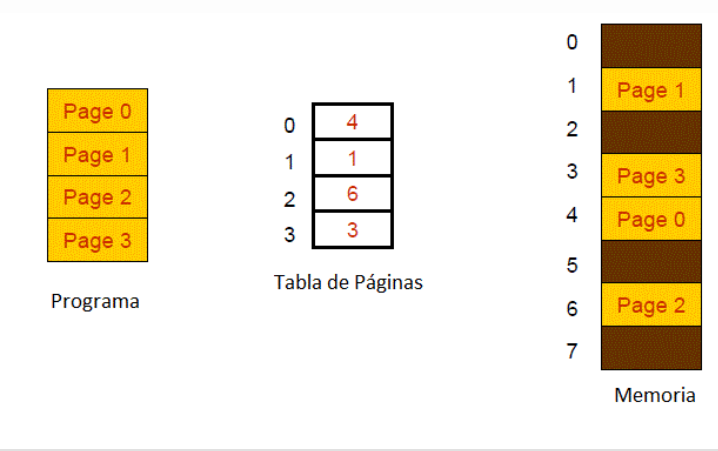
• La dirección de almacenamiento real se forma por la concatenación de la resolución de p (dirección inicio del marco que aloca la página) y d, donde p es el número de pagina y d es el desplazamiento

La transformación de direcciones lógicas a direcciones físicas en la paginación se realiza mediante la tabla de páginas. Cuando un proceso genera una dirección lógica, se divide en dos partes: el número de página y el desplazamiento dentro de esa página.

El número de página se usa como índice en la tabla de páginas para encontrar la entrada correspondiente.

La entrada de la tabla de páginas proporciona la dirección física de la página correspondiente en la memoria física.

Luego, se agrega el desplazamiento al final de la dirección física para encontrar la ubicación exacta en la memoria física.



d) En este esquema: ¿Se puede producir fragmentación (interna y/o externa)?

En la paginación, la fragmentación externa generalmente no es un problema, ya que las páginas se asignan de manera independiente y no se requieren bloques de memoria contiguos para alojar las páginas de un proceso. La paginación está diseñada específicamente para eliminar la fragmentación externa, ya que las páginas se asignan de manera dinámica a marcos de página en la memoria física según sea necesario.

Sin embargo, la fragmentación interna puede ser un problema potencial en la paginación. La fragmentación interna ocurre cuando un proceso no utiliza completamente la capacidad de una página. Por ejemplo, si una página tiene un tamaño de 4 KB y un proceso solo utiliza 3.5 KB de ese espacio, se desperdician 0.5 KB de memoria en esa página debido a la fragmentación interna.

La fragmentación interna en la paginación es generalmente menos problemática que la fragmentación externa en otros esquemas de gestión de memoria, ya que el desperdicio se limita al tamaño de página y no afecta la memoria en general. Además, la fragmentación interna se considera un compromiso aceptable en la paginación, dada su eficiencia en la gestión de la memoria y la eliminación de la fragmentación externa.

En resumen, en la paginación, la fragmentación externa no es un problema, y aunque puede haber fragmentación interna, generalmente se considera menos perjudicial que en otros esquemas de gestión de memoria, debido a la eficiencia global de la paginación.

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

Similitudes:

Gestión de memoria fragmentada: Tanto la paginación como las particiones fijas se utilizan para abordar el problema de la fragmentación en la administración de la memoria.

Protección y aislamiento de procesos: Ambos métodos permiten que múltiples procesos se ejecuten simultáneamente y se aíslen unos de otros en términos de memoria, lo que garantiza la protección y la seguridad de los procesos.

Asignación de memoria dinámica: En ambos enfoques, la asignación de memoria puede ser dinámica, lo que significa que se pueden asignar y liberar regiones de memoria según sea necesario.

Diferencias:

Tamaño de las divisiones de memoria:

Paginación: En la paginación, la memoria se divide en páginas de tamaño fijo y las páginas se asignan dinámicamente a marcos de memoria en la memoria física. El tamaño de las páginas es uniforme.

Particiones fijas: En la técnica de particiones fijas, la memoria se divide en particiones de tamaño fijo al principio del sistema. Cada partición puede albergar un proceso y el tamaño de las particiones se establece previamente.

Fragmentación interna y externa:

Paginación: La paginación elimina la fragmentación externa al no requerir bloques contiguos de memoria para procesos. Sin embargo, puede haber fragmentación interna en las páginas cuando un proceso no llena completamente una página.

Particiones fijas: Las particiones fijas pueden sufrir tanto de fragmentación interna como de fragmentación externa. La fragmentación interna ocurre cuando un proceso no utiliza completamente su partición asignada, y la fragmentación externa se produce cuando las particiones se desperdician debido a la falta de flexibilidad en el tamaño.

Flexibilidad:

Paginación: La paginación es más flexible en términos de asignación de memoria, ya que no requiere que los procesos se ajusten a particiones de tamaño fijo. Esto permite una mayor eficiencia en el uso de la memoria.

Particiones fijas: Las particiones fijas son menos flexibles, ya que los procesos deben adaptarse al tamaño de las particiones asignadas. Esto puede llevar a una menor eficiencia en la utilización de la memoria.

En resumen, tanto la paginación como las particiones fijas son técnicas para administrar la memoria en sistemas operativos, pero difieren en la forma en que dividen la memoria y abordan la fragmentación. La paginación ofrece una mayor flexibilidad y elimina la fragmentación externa, mientras que las particiones fijas pueden sufrir de fragmentación interna y externa debido a la asignación de bloques de tamaño fijo.

7.- Suponga un sistema donde la memoria es administrada mediante la técnica de paginación, y donde:

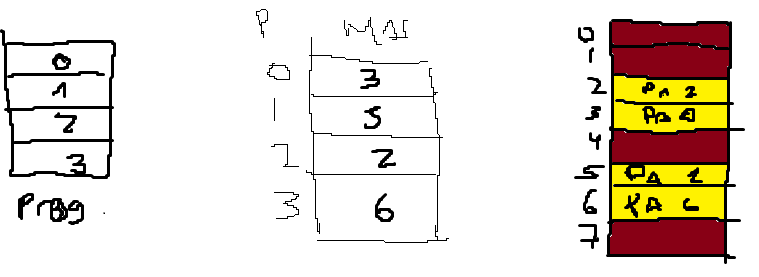
El tamaño de la página es de 512 bytes

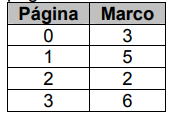
Cada dirección de memoria referencia 1 byte.

Los marcos en memoria principal de encuentran desde la dirección física 0.

Suponga además un proceso con un tamaño 2000 bytes y con la siguiente tabla de páginas:

a) Realice los gráficos necesarios (de la memoria, proceso y tabla de páginas) en el que reflejen el estado descrito.





marco inicio-fin

|  |  |
| --- | --- |
| 0 | 0-511 |
| 1 | 512-1023 |
| 2 | 1024-1535 |
| 3 | 1536-2047 |
| 4 | 2048-2559 |
| 5 | 2560-3071 |
| 6 | 3072-3583 |

b) Indicar si las siguientes direcciones lógicas son correctas y en caso afirmativo indicar la dirección física a la que corresponden:

i) 35 ii) 512 iii) 2051 NO valido iv) 0 v) 1325 vi) 602

i) 35/512=0 -> pag 0, marco 3... -> 35 mod 512= 35 -> 1536+35=**1571 direccion fisica.**

ii) 512/512=1 -> 1, marco 5. 512 mod 512=0 -> 2560+0 = **2560 dir fisica**

c) Indicar, en caso de ser posible, las direcciones lógicas del proceso que se corresponden si las siguientes direcciones físicas:

i) 509 ii) 1500 iii) 0 iv) 3215 v) 1024 vi) 2000

Al revés:

1024 div 512= 2 -> marco 2, pag 2 -> 1024 mod 512= 0 -> 0+(2\*512)= 1024 **dir virtual.**

**2560 div 512= 5 -> marco 5, pag 1 -> 2560 mod 512=0+512 = 512 dir virtual**

**1500 div 512= 2 -> marco 2, pag 2, 1500 mod 512 476 = 476+(num pag\* tamano=2\*512)=1500 dir virtual**

**3215 div 512= 6(marco)(pag 3) 3215 mod 512= 143 (desplaza) dir fisica= 3\*512 + 143=**

d) ¿Indique, en caso de que se produzca, la fragmentación (interna y/o externa)?

8)

Considere un espacio lógico de 8 paginas de 1024 bytes cada una, mapeadas en una memoria física de 32 marcos. a) ¿Cuantos bits son necesarios para representar una dirección lógica? b) ¿Cuantos bits son necesarios para representar una dirección física?

8\*1024=8192=2^13= **13 bits!**

**8 paginas= 2^3 + 1024 bits(2^10)=2^3+2^10=2^13**

**32\*1024= 32.768=2^15=15 biits!**

**2^5+ 2^10**

9)

Segmentación a) Explique como 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)?

a) Funcionamiento del Método de Asignación de Memoria por Segmentación:

En el método de asignación de memoria por segmentación, la memoria se divide lógicamente en segmentos, donde cada segmento es una unidad lógica relacionada con una porción del programa. Por ejemplo, puede haber segmentos para el código, datos, pila, etc. Cada segmento tiene un nombre y una longitud, y los programas se dividen en estos segmentos.

Cuando un programa solicita memoria, el sistema operativo asigna segmentos de memoria según las necesidades del programa. Cada segmento tiene un tamaño variable y no necesariamente contiguo en la memoria física.

b) Estructuras Adicionales del SO para Implementar Segmentación:

Para implementar la segmentación, el sistema operativo debe mantener información adicional sobre los segmentos. Se requieren tablas de segmentos que contengan información sobre la base (inicio) y el límite (tamaño) de cada segmento. Además, podría haber tablas de páginas dentro de cada segmento si se utiliza paginación dentro de los segmentos.

c) Transformación de Direcciones Lógicas a Físicas en Segmentación:

En el esquema de segmentación, las direcciones lógicas se componen de dos partes: el número de segmento y el desplazamiento dentro del segmento. La transformación de direcciones lógicas a físicas implica buscar la base del segmento en la tabla de segmentos y sumarle el desplazamiento.

Las direcciones logicas constan de dos partes → un numero de segmento s y un desplazamiento d dentro del segmento (0 ⟨ d ⟨ lımite)

Direccion fısica=Base del segmento+Desplazamiento

• La t´ecnica de paginaci´on intenta alocar la mayor cantidad de p´aginas necesarias posibles

• Cada vez que hay que alocar una p´agina en un marco, se produce un fallo de p´agina (page fault) → hard page fault

• ¿Qu´e sucede si es necesario alocar una p´agina y ya no hay espacio disponible? • Se debe seleccionar una p´agina v´ıctima, para lo cual existen diversos algoritmos

• ¿Cu´al es el mejor algoritmo?:

• El que seleccione como p´agina v´ıctima aquella que no vaya a ser referenciada en un futuro pr´oximo

• La mayoria de los algoritmos predicen el comportamiento futuro mirando el comportamiento pasado

• Selecciona la p´agina cuyo pr´oxima referencia se encuentra m´as lejana a la actual

• Imposible de implementar → no se conoce los futuros eventos

10)

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

Similitudes entre Segmentación y Particiones Dinámicas:

División de la Memoria: Tanto la segmentación como las particiones dinámicas dividen la memoria en unidades más pequeñas para asignar a los procesos.

Uso Eficiente de la Memoria: Ambas técnicas buscan utilizar eficientemente la memoria al asignar bloques o segmentos de memoria según las necesidades de los procesos.

Asignación Dinámica: En ambas técnicas, la asignación de memoria puede realizarse de manera dinámica, es decir, a medida que los procesos la solicitan.

Diferencias entre Segmentación y Particiones Dinámicas:

Estructura de Asignación:

Segmentación: La memoria se divide en segmentos lógicos que pueden tener tamaños variables y no necesariamente contiguos. Cada segmento corresponde a una unidad lógica del programa (código, datos, pila, etc.).

Particiones Dinámicas: La memoria se divide en particiones más pequeñas, pero estas particiones pueden ser contiguas o no. Cada partición puede asignarse a un solo proceso.

Flexibilidad de Tamaño:

Segmentación: Permite mayor flexibilidad en el tamaño de los segmentos, lo que puede adaptarse mejor a la estructura lógica del programa.

Particiones Dinámicas: Las particiones suelen tener tamaños fijos, lo que puede llevar a una asignación menos eficiente si los procesos no encajan exactamente en las particiones disponibles.

Fragmentación:

Segmentación: Puede haber fragmentación externa entre segmentos asignados, y también fragmentación interna dentro de un segmento.

Particiones Dinámicas: Puede haber fragmentación interna (en el último bloque asignado a un proceso) y fragmentación externa (bloques libres no contiguos).

Tablas de Asignación:

Segmentación: Requiere tablas de segmentos que contienen información sobre la base y el límite de cada segmento.

Particiones Dinámicas: No necesariamente requiere tablas específicas, pero puede usar una tabla de asignación para mantener un registro de las particiones ocupadas y libres.

Adaptabilidad a la Carga de Trabajo:

Segmentación: Puede adaptarse mejor a la carga de trabajo variable de los procesos, ya que los segmentos pueden crecer o disminuir según sea necesario.

Particiones Dinámicas: Puede tener problemas con la fragmentación externa si la carga de trabajo varía mucho y las particiones no se pueden reorganizar fácilmente.

En resumen, ambas técnicas comparten el objetivo de asignar y utilizar eficientemente la memoria, pero difieren en la estructura y en la forma en que manejan la asignación de memoria a los procesos. La segmentación tiende a ser más flexible en términos de tamaño y estructura lógica, mientras que las particiones dinámicas pueden tener un enfoque más simple pero pueden sufrir más de fragmentación.

11)

Cite similitudes y diferencias entre la técnica de paginación y segmentación

Similitudes entre Paginación y Segmentación:

División Lógica de la Memoria: Ambas técnicas dividen la memoria lógica en unidades más pequeñas para facilitar la administración y asignación de memoria.

Soporte para Espacios de Direcciones Virtuales:

Paginación: Permite la creación de espacios de direcciones virtuales mediante la asignación de páginas.

Segmentación: Permite la creación de espacios de direcciones virtuales mediante la asignación de segmentos.

Protección de Memoria:

Paginación: Puede proporcionar protección de memoria a nivel de página.

Segmentación: Puede proporcionar protección de memoria a nivel de segmento.

Administración de Memoria de Manera No Contigua:

Paginación: Permite que las páginas de un proceso se almacenen de manera no contigua en la memoria física.

Segmentación: Permite que los segmentos de un proceso se almacenen de manera no contigua en la memoria física.

Diferencias entre Paginación y Segmentación:

Estructura:

Paginación: La memoria se divide en bloques fijos llamados "páginas". Toda la memoria se trata como un conjunto homogéneo de páginas.

Segmentación: La memoria se divide en unidades lógicas más grandes llamadas "segmentos". Cada segmento tiene un nombre y puede tener un tamaño variable.

Tamaño de las Unidades:

Paginación: Todas las páginas tienen el mismo tamaño fijo.

Segmentación: Los segmentos pueden tener tamaños variables.

Manejo de Fragmentación:

Paginación: Puede haber fragmentación interna en las páginas (si no se utiliza completamente una página).

Segmentación: Puede haber fragmentación interna en los segmentos y fragmentación externa entre segmentos.

Tabla de Páginas vs. Tabla de Segmentos:

Paginación: Utiliza tablas de páginas para realizar la traducción de direcciones lógicas a físicas.

Segmentación: Utiliza tablas de segmentos para realizar la traducción de direcciones lógicas a físicas.

Simplicidad de Implementación:

Paginación: Tiende a ser más simple de implementar y manejar, especialmente para sistemas operativos y hardware.

Segmentación: Puede ser más compleja de implementar, especialmente cuando se manejan segmentos de tamaños variables.

Fragmentación Externa:

Paginación: No tiene fragmentación externa, ya que las páginas son de tamaño fijo y contiguas.

Segmentación: Puede tener fragmentación externa debido a la asignación no contigua de segmentos.

Ambas técnicas, paginación y segmentación, tienen ventajas y desventajas, y la elección entre ellas a menudo depende de los requisitos específicos del sistema y de la carga de trabajo esperada. Además, en muchos sistemas modernos, se utiliza una combinación de paginación y segmentación, conocida como paginación segmentada.

12)

A

13)

Memoria Virtual

a) Describa que beneficios introduce este esquema de administración de la memoria.

b) ¿En que se debe apoyar el SO para su implementación?

c) Al implementar está técnica utilizando paginación por demanda, las tablas de paginas de un proceso deben contar con información adicional además del marco donde se encuentra la página. ¿Cuál es está información? ¿ Porque es necesaria?

Memoria Virtual:

a) Beneficios de la Memoria Virtual: La memoria virtual es un esquema que proporciona al sistema operativo y a los programas la ilusión de que tienen más memoria física de la que realmente está disponible. Los beneficios clave incluyen:

Mayor Capacidad de Ejecución: Permite que los programas se ejecuten incluso si su tamaño total excede la cantidad de memoria física disponible. Esto es esencial para ejecutar aplicaciones más grandes.

Isolación de Procesos: Proporciona un entorno aislado para cada proceso, lo que significa que un proceso no puede acceder directamente a la memoria de otro proceso.

Multiprogramación Eficiente: Permite que varios programas se ejecuten simultáneamente sin necesidad de que todos los programas estén completamente cargados en la memoria física al mismo tiempo.

Mayor Utilización de la Memoria: Permite a los sistemas operativos y a los programas utilizar más eficientemente la memoria física al cargar solo las partes necesarias de un programa en la memoria en un momento dado.

b) Requisitos del SO para la Implementación de Memoria Virtual: Para implementar la memoria virtual, el sistema operativo debe proporcionar los siguientes elementos:

Unidad de Gestión de Memoria (MMU): Un componente hardware o software que realiza la traducción de direcciones lógicas a direcciones físicas, implementando la paginación o segmentación.

Gestión de Páginas o Segmentos: Un sistema para administrar las páginas o segmentos en la memoria física y en el almacenamiento secundario.

Algoritmos de Reemplazo de Páginas: Cuando se necesita cargar una página y no hay suficiente espacio en memoria, se deben utilizar algoritmos de reemplazo para determinar qué página reemplazar.

Algoritmos de Asignación de Páginas: Para decidir qué páginas cargar en memoria cuando se necesita espacio.

Gestión de Espacio en Disco: Para almacenar páginas o segmentos no presentes en la memoria en un almacenamiento secundario (disco).

c) Información Adicional en Tablas de Páginas para Paginación por Demanda: Cuando se implementa memoria virtual utilizando paginación por demanda, además de la entrada que indica el marco en el que se encuentra una página, la tabla de páginas de un proceso debe incluir información adicional, como:

Bit de Presente/Ausente: Indica si la página está presente en la memoria física o si se encuentra en el almacenamiento secundario.

Bit de Modificación (Dirty Bit): Indica si la página ha sido modificada desde que se cargó en la memoria. Esto es útil para decidir si es necesario escribir la página de nuevo en el almacenamiento secundario antes de reemplazarla.

Bit de Acceso: Indica si la página ha sido accedida recientemente. Ayuda en la implementación de algoritmos de reemplazo de páginas.

Estos bits adicionales son necesarios para que el sistema operativo pueda administrar eficientemente la paginación por demanda y garantizar que solo las páginas necesarias se carguen en 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 cando se produce un fallo de página.

los fallos de página están directamente relacionados con la paginación. La paginación es una técnica de administración de memoria virtual en la que la memoria lógica y la memoria física se dividen en bloques llamados "páginas", y estas páginas son las unidades básicas de transferencia entre el almacenamiento secundario y la memoria física. Cuando un programa intenta acceder a una página que no está presente en la memoria física, se produce un fallo de página.

Cuando un programa solicita el acceso a una dirección de memoria que no está en la memoria física, se produce una excepción de fallo de página. La Unidad de Gestión de Memoria (MMU) detecta esta excepción y señala al sistema operativo que se debe tomar una acción.

En el caso de la paginación por demanda, el sistema operativo no carga todas las páginas de un programa en la memoria física de una vez, sino que las carga a medida que se accede a ellas. Cuando se produce un fallo de página, el sistema operativo interviene, carga la página necesaria desde el almacenamiento secundario a la memoria física y luego permite que el programa continúe su ejecución.

Por lo tanto, la paginación y los fallos de página están intrínsecamente vinculados, ya que la paginación por demanda es una estrategia clave para gestionar eficientemente la memoria virtual y permitir que los programas se ejecuten en sistemas con limitaciones de memoria física.

a) ¿Cuándo se Producen los Fallos de Página? Los fallos de página ocurren cuando un programa intenta acceder a una página de memoria que no se encuentra actualmente en la memoria física. Es decir, cuando una dirección lógica apunta a una página que no está presente en la memoria RAM.

b) ¿Quién es Responsable de Detectar un Fallo de Página? La detección de un fallo de página es responsabilidad del hardware, específicamente de la Unidad de Gestión de Memoria (MMU). Cuando un programa intenta acceder a una dirección lógica que no está presente en la memoria física, se genera una interrupción conocida como una excepción de fallo de página. El hardware detecta esta excepción y la comunica al sistema operativo.

c) Acciones del SO Cuando se Produce un Fallo de Página: Cuando se produce un fallo de página, el sistema operativo toma varias acciones para manejar la situación:

Excepción de Fallo de Página:

El hardware genera una excepción de fallo de página cuando se detecta que la página necesaria no está presente en la memoria física.

Interrupción del SO:

El hardware informa al sistema operativo sobre el fallo de página a través de una interrupción. La interrupción proporciona información sobre la dirección lógica que causó el fallo y posiblemente otros detalles.

Identificación de la Página Requerida:

El sistema operativo identifica la página requerida que no está en la memoria física.

Manejo del Fallo de Página:

El SO busca la página en el almacenamiento secundario (generalmente en disco).

Si la página no está en el almacenamiento secundario, puede ser necesario cargarla desde el sistema de archivos o incluso desde recursos de red.

Actualización de las Tablas de Páginas:

Una vez que la página está en la memoria física, el sistema operativo actualiza las tablas de páginas para reflejar la nueva ubicación de la página.

Esto puede incluir la actualización de la tabla de páginas del proceso y otros registros necesarios.

Reinicio de la Instrucción:

La instrucción que causó el fallo de página se reinicia para que ahora pueda ejecutarse con la página presente en la memoria física.

Continuación de la Ejecución del Programa:

Después de manejar el fallo de página, el sistema operativo permite que el programa continúe su ejecución normalmente.

El manejo eficiente de los fallos de página es esencial para garantizar que los programas puedan ejecutarse sin problemas, incluso cuando la memoria física no es suficiente para contener todo el espacio de direcciones de un programa.

15)

Direcciones: a) Si se dispone de una 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 pagina es de 512Kb. ¿Cuál es el número máximo de paginas que puede tener un proceso? iii) Si el tamaño de pagina es de 512Kb. y se disponen de 256 Mb. de memoria real ¿Cuál es el número de marcos que puede haber? iv) Si se utilizaran 2 Kb. para cada entrada en la tabla de páginas de un proceso: ¿Cuál seria el tamaño máximo de la tabla de páginas de cada proceso?

a) Espacio de Direcciones Virtuales de 32 bits:

i) Tamaño Máximo de un Proceso:

En un espacio de direcciones virtuales de 32 bits, cada dirección puede representar 232232 bytes.

El tamaño máximo de un proceso es 232232 bytes o 4 GB.

ii) Número Máximo de Páginas por Proceso con Tamaño de Página de 512 KB:

Cada página es de 512 KB, por lo que el número máximo de páginas por proceso es 232512 KB512 KB232 .

232512 KB=8192 paˊginas512 KB232 =8192 paˊginas

iii) Número de Marcos con Tamaño de Página de 512 KB y 256 MB de Memoria Real:

El número de marcos de página en memoria real es igual a la capacidad total de la memoria real dividida por el tamaño de la página.

256 MB512 KB=512512 KB256 MB =512 marcos.

iv) Tamaño Máximo de la Tabla de Páginas con Entradas de 2 KB cada una:

Cada entrada en la tabla de páginas es para una página, y cada entrada es de 2 KB.

El tamaño máximo de la tabla de páginas es igual al número máximo de páginas por proceso multiplicado por el tamaño de cada entrada.

8192 paˊginas×2 KB/entrada=16 MB8192 paˊginas×2 KB/entrada=16 MB

Por lo tanto:

i) Tamaño máximo de un proceso: 4 GB.

ii) Número máximo de páginas por proceso: 8192 páginas.

iii) Número de marcos con tamaño de página de 512 KB y 256 MB de memoria real: 512 marcos.

iv) Tamaño máximo de la tabla de páginas con entradas de 2 KB cada una: 16 MB.

16)