



Nombre: Dennys Alexander Pucha Carrera

Paralelo: 4to "A"

Fecha: 01/08/2023

Asignatura: Sistemas Operativos

Docente: Ing. Hernán Leonardo Torres Carrión M.Sc.

ENSAYO Nº 11

1. Tema

Paginación y Segmentación en la Memoria Principal

2. Antecedentes

En el presente ensayo, se tratará temas como la gestión de memoria en sistemas operativos y arquitecturas de hardware. El enfoque se centrará en dos técnicas fundamentales: la paginación y la segmentación. Estas estrategias de manejo de memoria son piezas clave para lograr una eficiente utilización de los recursos y un óptimo rendimiento en la ejecución de programas.

El objetivo principal será analizar cómo estas técnicas influyen en la administración de la memoria, evaluando sus ventajas y desventajas. Se examinará cómo la paginación mejora la utilización global de la memoria y cómo la segmentación brinda mayor flexibilidad para programas con requisitos de memoria variables.

Durante el recorrido de este ensayo, se presentará información cómo la paginación y la segmentación han sido fundamentales para permitir la implementación de la memoria virtual en sistemas operativos modernos, lo que ha llevado a una multitarea más eficiente y a una experiencia del usuario mejorada.

Para un estudiante de computación es de suma importancia el comprender como es que la segmentación y paginación de la memoria ayuda a tener una multitarea mucho más eficiente ya que mediante esta comprensión se puede identificar el trabajo interno sobre los programas y como estos se optimizan para un mejor funcionamiento del sistema.

3. Descripción

Para entender que es la paginación y segmentación de la memoria y lo que involucra es necesario comprender que es cada concepto, para esto se tienen las siguientes definiciones empezando por paginación se puede decir que es:

Es una técnica de manejo de memoria, en la cual el espacio de memoria se divide en secciones físicas de igual tamaño, denominadas marcos de página. Los programas se dividen en unidades lógicas denominadas páginas, que tienen el mismo tamaño que los marcos de páginas. [1]

Lo que dice esta definición concreta es que la paginación es una forma de memoria virtual que permite que un programa se divida en bloques o unidades lógicas llamadas "páginas". Cada página tiene el mismo tamaño, que generalmente es igual al tamaño de los marcos de página en la memoria física. Los marcos de página son regiones contiguas de memoria física que también tienen el mismo tamaño que las páginas.

En lugar de cargar todo el programa en la memoria principal, lo cual puede ser costoso en



términos de recursos, la paginación permite que solo las páginas requeridas en un momento dado se carguen en la memoria física.

Cuando una página es necesaria para la ejecución de un programa, se carga desde el almacenamiento secundario (como el disco duro) a un marco de página libre en la memoria principal. Si la página ya está en la memoria principal, no es necesario cargarla nuevamente.

Para mantener un seguimiento de la correspondencia entre las páginas lógicas y los marcos de página físicos, se utiliza una tabla llamada "tabla de páginas" (page table). Esta tabla almacena las direcciones físicas correspondientes a las direcciones lógicas de las páginas del programa.

En cuanto a la segmentación se puede decir que es:

Es un esquema de manejo de memoria mediante el cual la estructura del programa refleja su división lógica, llevándose a cabo una agrupación lógica de la información en bloques de tamaño variable denominados segmentos, es decir que los segmentos pueden ser de distintos tamaños, incluso de forma dinámica.[1]

A partir de esta definición se puede decir que la segmentación es una forma de memoria virtual en la que el espacio de memoria se divide en bloques de tamaño variable llamados "segmentos". Cada segmento corresponde a una unidad lógica o funcional del programa, como el código del programa, los datos globales, la pila, etc. Estos segmentos pueden tener tamaños diferentes y no necesariamente están ubicados en áreas contiguas de memoria física.

A diferencia de la paginación, que divide el programa en unidades de tamaño fijo (páginas) y los almacena en marcos de página también de tamaño fijo en la memoria física, la segmentación permite que los segmentos tengan tamaños diferentes y pueden crecer o disminuir dinámicamente según las necesidades del programa.

A continuación, se presenta una tabla de ventajas y desventajas tanto de paginación como de segmentación:

Paginación:

Ventajas	Desventajas
Mayor utilización de la memoria	Puede haber un pequeño costo adicional en tiempo debido a la traducción de direcciones y el acceso a la tabla de páginas
No hay fragmentación externa	Puede haber cierta fragmentación interna debido a los tamaños fijos de las páginas y a la asignación de múltiples páginas para procesos que no las necesitan completamente
Facilita la gestión de la memoria virtual	Requiere el uso de una tabla de páginas que debe mantenerse actualizada, lo que puede agregar complejidad
Permite gestionar de manera eficiente el espacio de memoria cuando se ejecutan múltiples procesos [2]	



Segmentación:

Ventajas	Desventajas
Mayor flexibilidad en el manejo de memoria	Puede haber fragmentación interna y externa, lo que puede desperdiciar memoria
Permite que los segmentos crezcan o disminuyan dinámicamente según las necesidades del programa	Puede haber un costo adicional en tiempo debido a la traducción de direcciones y al acceso a la tabla de segmentos
Facilita la organización lógica del programa	Requiere el uso de una tabla de segmentos que debe mantenerse actualizada, lo que puede agregar complejidad
Es útil para programas con requisitos de memoria variables durante su ejecución [2]	

A continuación, se presenta una descripción del funcionamiento tanto de la paginación como de la segmentación:

Funcionamiento de la Paginación:

1. División en páginas: El programa se divide en unidades lógicas de tamaño fijo llamadas "páginas". Cada página tiene el mismo tamaño que los "marcos de página" en la memoria física.
2. Tabla de páginas: Se utiliza una tabla de páginas (page table) para realizar la traducción entre las direcciones lógicas de las páginas y las direcciones físicas de los marcos de página. Esta tabla contiene entradas que asocian cada número de página lógica con el número de marco de página física correspondiente.
3. Memoria virtual y física: El programa utiliza direcciones lógicas que se dividen en un número de página y un desplazamiento dentro de esa página. El hardware se encarga de traducir las direcciones lógicas a direcciones físicas utilizando la tabla de páginas.
4. Gestión de memoria: Cuando una página es necesaria para la ejecución de un programa y no se encuentra en la memoria física, se produce un "fallo de página" (page fault). En ese caso, la página se carga desde el almacenamiento secundario (disco duro) a un marco de página libre en la memoria principal. Si el marco de página seleccionado ya estaba ocupado, se utiliza un algoritmo de reemplazo para decidir qué página reemplazar en la memoria.
5. Acceso a memoria: Una vez que una página está en la memoria física, el acceso a los datos o instrucciones almacenadas en esa página es directo, ya que el hardware se encarga de traducir las direcciones lógicas a direcciones físicas.[3]



Funcionamiento de la Segmentación:

1. División en segmentos: El programa se divide en bloques de tamaño variable llamados "segmentos". Cada segmento corresponde a una unidad lógica o funcional del programa, como el código del programa, los datos globales, la pila, etc.
2. Tabla de segmentos: Se utiliza una tabla de segmentos (segment table) para realizar la traducción entre las direcciones lógicas de los segmentos y las direcciones físicas en la memoria principal. Esta tabla contiene entradas que asocian cada número de segmento lógico con la dirección base y el límite del segmento en la memoria física.
3. Memoria virtual y física: El programa utiliza direcciones lógicas que se dividen en un número de segmento y un desplazamiento dentro de ese segmento. El hardware se encarga de traducir las direcciones lógicas a direcciones físicas utilizando la tabla de segmentos.
4. Gestión de memoria: Los segmentos pueden crecer o disminuir dinámicamente según las necesidades del programa. Cuando un segmento crece y no hay suficiente espacio contiguo en la memoria física, puede ocurrir fragmentación externa, lo que requiere de técnicas de administración de memoria para resolverlo.
5. Acceso a memoria: Una vez que un segmento está en la memoria física, el acceso a los datos o instrucciones almacenadas en ese segmento es directo, ya que el hardware se encarga de traducir las direcciones lógicas a direcciones físicas. [4]

La paginación y la segmentación son técnicas de manejo de memoria que buscan optimizar el uso de los recursos de la computadora y mejorar el rendimiento de los programas. Cada enfoque aborda diferentes aspectos de la gestión de memoria y tiene sus propias ventajas y desventajas.

La paginación permite dividir el programa en bloques de tamaño fijo llamados páginas, lo que facilita la administración de la memoria y permite una mayor utilización de los recursos. Además, la paginación es eficiente en términos de fragmentación externa y permite que múltiples procesos se ejecuten en la memoria al mismo tiempo. Sin embargo, puede haber cierta fragmentación interna debido a los tamaños fijos de las páginas y puede requerir un tiempo adicional para traducir las direcciones lógicas a direcciones físicas utilizando la tabla de páginas.

Por otro lado, la segmentación divide el programa en bloques de tamaño variable llamados segmentos, lo que brinda una mayor flexibilidad en el manejo de la memoria y permite que los segmentos crezcan o disminuyan dinámicamente según las necesidades del programa. Esto es especialmente útil para programas con requisitos de memoria variables durante su ejecución. Sin embargo, la segmentación puede llevar a fragmentación interna y externa, lo que puede desperdiciar memoria y requiere una tabla de segmentos para traducir las direcciones lógicas a direcciones físicas, lo que puede tener un costo adicional en tiempo.



4. Conclusiones

- Paginación y segmentación son técnicas esenciales en la gestión de memoria de sistemas operativos y arquitecturas de hardware. Su implementación adecuada puede marcar la diferencia entre un sistema eficiente y uno con problemas de uso de memoria y rendimiento. Comprender cómo funcionan estas técnicas permite optimizar el uso de la memoria, asegurando una asignación adecuada de recursos y evitando la sobrecarga innecesaria.
- La elección entre paginación y segmentación depende de las necesidades y características del programa y del sistema en cuestión. Ambas técnicas tienen ventajas y desventajas que deben ser consideradas al tomar decisiones sobre la gestión de memoria. La segmentación puede ser más adecuada para programas con requisitos de memoria variables, mientras que la paginación puede ser preferible para mejorar la utilización global de la memoria.
- La paginación y la segmentación han sido fundamentales para permitir la implementación de memoria virtual en sistemas operativos modernos. La memoria virtual permite que los programas se ejecuten como si tuvieran acceso a toda la memoria disponible, aunque en realidad solo una parte se carga en la memoria física en un momento dado. Esto brinda mayor flexibilidad y eficiencia en la administración de los recursos de memoria.

5. Bibliografía

- [1] P. Guaman, "Paginación y Segmentación de un Sistema Operativo," [En línea]. Disponible en: <http://paoguaman.blogspot.com/2016/02/paginacion-y-segmentacion-de-un-sistema.html>. [Accedido:01-08-2023].
- [2] A. Corral, "TEMA 3. GESTIÓN DE MEMORIA," Universidad de Almería, [En línea]. Disponible en: https://w3.ual.es/~acorral/DSO/Tema_3.pdf. [Accedido:01-08-2023].
- [3] Universidad Autónoma del Estado de Hidalgo, "Sistemas de Paginación Segmentación," [En línea]. Disponible en: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro26/sistemas_de_paginacin_segmentacin.html. [Accedido:01-08-2023].
- [4] R. García Carmona, "GESTIÓN DE MEMORIA," Universidad CEU San Pablo, [En línea]. Disponible en: https://repositorioinstitucional.ceu.es/bitstream/10637/9342/6/SOS-6-Gestion_Memoria_RGarciaCarmona_2014.pdf. [Accedido:01-08-2023].