

UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORIA ACADEMICA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
CARRERA INGENIERÍA INFORMÁTICA

Tarea No 1

MODALIDAD ESCOGIDA: TAREA

TAREA #1 PARA EL CURSO
DE Sistemas Operativos

PABLO ANDRÉ VALENCIANO BLANCO

1-1572-0043

CENTRO UNIVERSITARIO DE HEREDIA

PAC: 2024-2

CIUDAD: HEREDIA

Contenido

Introducción.....	3
Parte 1. ¿Cómo se administra la memoria en los Sistemas Operativos de Windows, de Mainframe y de Android?	4
Windows	5
Mainframe	5
Android	7
Parte 2. ¿Qué es paginación y qué es segmentación?	8
Parte 3. ¿Cómo se administra la memoria en el procesador Intel Core?	11
Parte 4. Cuadro comparativo de los algoritmos de reemplazo de páginas ventajas y desventajas.	13
Conclusión.....	15
Referencias	16

Índice de Figuras

Figura 1.....	4
Figura 2.....	7
Figura 3.....	8
Figura 4.....	12

Índice de Tablas

Tabla 1.....	9
Tabla 2.....	13

Introducción

El siguiente documento dará por presentado la primera tarea del curso de Sistemas Operativos, el cual va a ser trabajado por el estudiante Pablo Valenciano, el proyecto va a tener como énfasis los siguientes 2 temas del curso: 2.1 Memoria y 2.2 Archivos.

Para el enfoque se va hacer un análisis similar y más profundo de como la memoria se administra para ciertos sistemas operativos y como se relacionan con las entradas y salidas del sistema y conceptos de como la memoria realiza las funcionalidad y conceptos paginación y segmentación.

Los sistemas operativos por si solos no son capaces de contener el hardware, de ahí que los procesadores de distintas marcas dan acceso al sistema operativo de usar los dispositivos los procesadores, con las memorias caches que estas poseen y las memorias RAM que se conectan y la ROM que esta ya programada para hacer funcionar el dispositivo, el sistema operativo es cualquier opción “bootable” que permite al usuario ingresar comandos que interactúen con las Entradas y Salidas.

Mientras tanto los archivos que serán almacenados y ejecutados por las memorias, siendo el sistema operativo la interfaz de los procesos, estos deben ser controlados con algoritmos que manejen de forma eficiente y nos permiten trabajar en diferentes hilos cada uno de los procesos sin bloquear los procesos esenciales y no esenciales de la máquina, protegiendo así el uso de los recursos.

Para este documento se dividirá cada sección, según lo estipulado por la tarea:

- a) Administración de la memoria en los SO: Windows, Mainframe y Android.
- b) Conceptos de Paginación y Segmentación.
- c) La forma como se administra la memoria del Procesador Intel Core, niveles de cache.
- d) Algoritmos de remplazo de páginas, lo que comparten y diferencian, uso de cuadro comparativo.

Al finalizar se darán las respectivas conclusiones del trabajo realizado, para cada una de las partes previamente mencionadas.

Parte 1. ¿Cómo se administra la memoria en los Sistemas

Operativos de Windows, de Mainframe y de Android?

Según la Editorial Etece (2013), se define el sistema operativo como el software que coordina y dirige todos los servicios y aplicaciones que utiliza el usuario en una computadora, por eso es el más importante y fundamental.

Los sistemas operativos, también llamados núcleos o kernels, suelen ejecutarse de manera privilegiada respecto al resto del software, sin permitir que un programa cualquiera realice cambios de importancia sobre él que puedan comprometer su funcionamiento.

El sistema operativo es el protocolo básico de operatividad del computador, que coordina todas sus demás funciones de comunicaciones, de procesamiento, de interfaz con el usuario.

Los sistemas operativos se componen según lo visto por la figura 1.

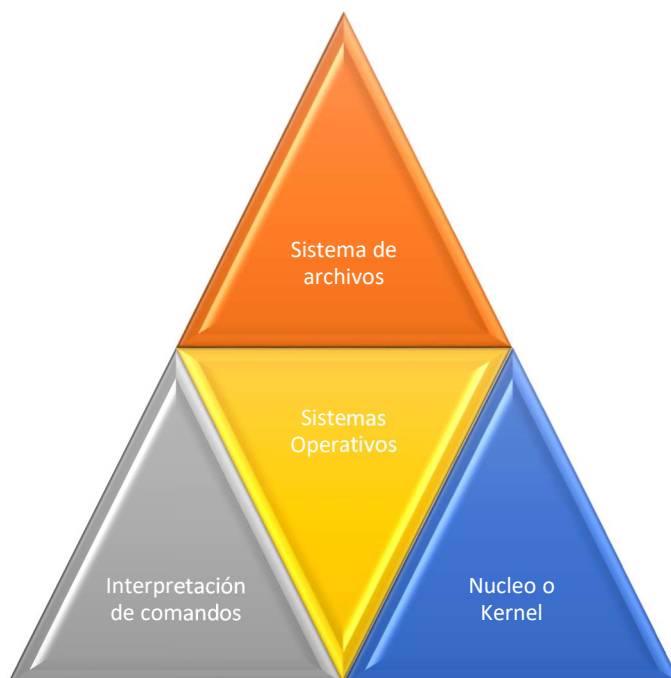


Figura 1. Componentes requeridos para cualquier sistema operativo

Windows

Windows el Sistema operativo, más usado en el mundo según las estadísticas brindadas por la página oficial de statistics (2023), donde para el año 2023, domina el mercado con un 63%. La forma como administra la memoria a día de hoy, con los últimos cambios de versiones lo realiza por medio del administrador de memoria de Windows Kernel-Mode.

El componente administrador de memoria en modo kernel de Windows administra la memoria física para el sistema operativo. Esta memoria se basa principalmente en la forma de memoria de acceso aleatorio (RAM).

El administrador de memoria administra la memoria realizando las siguientes tareas principales:

- Administrar la asignación y desasignación de memoria de forma virtual y dinámica.
- Compatibilidad con los conceptos de archivos asignados a memoria, memoria compartida y copia en escritura.

Las rutinas que proporcionan una interfaz directa al administrador de memoria suelen tener el prefijo "Mm"; por ejemplo, MmGetPhysicalAddress.

Mainframe

Según Brutti (2023), El mainframe fue creado para desarrollar un sistema computarizado para las transacciones. Incluso hoy, más de 60 años después de su creación, este es un ordenador de alto rendimiento bastante robusto, dinámico y eficaz a la hora de manejar enormes cantidades de información.

Este ordenador es fundamental para aplicaciones críticas, como la creación de bases de datos y almacenes de datos, especialmente en el sector financiero. Todo gracias a su potencia y a sus estándares de seguridad.

Es vital para:

- Procesamiento y análisis de datos.
- Procesamiento de transacciones.

- Registros a gran escala.
- Censos.
- Estudios estadísticos del consumidor.

Entre otros usos.

La forma en cómo se administra la memoria en los mainframes es colocando el sistema operativo dentro de la RAM, la cual para poder llevar a cabo sus procesos y la escalabilidad de más programas, se debe ampliar la memoria volátil teniendo que hacer que estos sistemas sean más robustos, pero a la misma vez más rápidos, ya que todos los datos se direccionaran al mismo sistema operativo y no hay necesidad de tener que pasar por una ROM más grande y más lenta para poder tener acceso.

Los sistemas operativos que se manejan bajo esta forma de trabajar, son en su mayoría aquellos que almacenan y consultan bases de datos de alta confidencialidad, como uso en salud, financieros y gubernamentales. Internacionalmente, de las más conocidas son aquellas basadas en Linux, los cuales trabajan en servidores y clústeres, tales como Redhat, Centos y Fedora, también hay empresas como IBM enfocadas en tener edificios de gran tamaño con miles de computadores con el sistema operativo propio IBM z/OS, ofreciendo servicios de Cloud.

La versión actual ofrecida es la IBM z/OS 3.1 según la página oficial de IBM, ofrece las siguientes características y ventajas sobre sus competidos:

- a) Escalar el valor de los datos e impulsar la transformación digital impulsada por la IA y la automatización inteligente
- b) Proteger y prosperar con unas excepcionales prestaciones de ciberresiliencia
- c) Gestionar y optimizar la infraestructura tecnológica con interfaces intuitivas
- d) Crear nuevas aplicaciones y modernizar las existentes con una estrategia flexible de nube híbrida

Para más información del IBM z/OS 3.1 visitar el siguiente [enlace](#).

Android

Android es un Sistema operativo basado en Linux con el kernel 2.6.x, simplificada para manejar la mayoría de tareas. Utiliza bibliotecas nativas en C abiertas. Todas las operaciones básicas del sistema operativo como de E / S, gestión de memoria, y así sucesivamente, son manejados por el de kernel de Linux. De esta forma indica en la página Zospic.

La gestión de memoria en Android asegura la respuesta de la aplicación, deteniendo y matando a los procesos que obstaculizan la fluidez y libera recursos para las aplicaciones de mayor prioridad. Cada aplicación Android se ejecuta en un proceso independiente dentro de su propia instancia de Dalvik, renunciando a toda responsabilidad de la memoria y la gestión de procesos.

Dalvik y el entorno de ejecución de Android se posicionan en la parte superior de un núcleo de Linux que se encarga de la interacción de bajo nivel del hardware, incluyendo los drivers y la gestión de memoria, mientras que el conjunto de API proporciona acceso a todos los servicios de bajo nivel, características y hardware. Dalvik Virtual Machine es una máquina virtual basada en registros que ha sido optimizada para asegurar que un dispositivo puede ejecutar múltiples instancias de manera eficiente. Se basa en el kernel de Linux para la gestión de memoria de bajo nivel.

Los programas reclaman los recursos del dispositivo según sea la prioridad acorde a la figura 2.

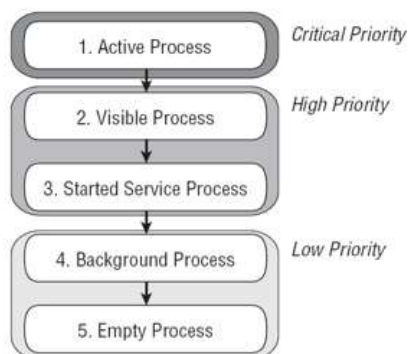


Figura 2. Orden de Prioridades para el Sistema Operativo Android

Parte 2. ¿Qué es paginación y qué es segmentación?

Paginación y segmentación son técnicas de gestión de la memoria, la importancia de manejar se deba a las siguientes razones:

- 1) Jerarquía de memoria: Cada sistema debe conocer la forma como la memoria administrada y donde debe almacenar los datos que en estas se ocupan, según sea su frecuencia, el programa que la esté usando y los servicios y programas, que están realizando en segundo plano, donde el sistema operativo, debe buscar la manera de intentar lo más eficiente, prediciendo cual va a ser la siguiente instrucción a ser ejecutada. La memoria debe estar segmentada, según cantidad y velocidad.

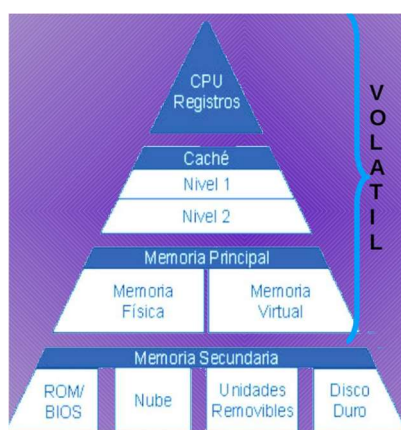


Figura 3. Jerarquía de la Memoria.

- 2) Reubicación de los programas: La memoria debe tener la capacidad de según como se estén realizando las tareas, ubicar de forma relativa los procesos dándole tres direcciones a los procesos que se estén manejando, y saber cuándo pasar de una jerarquía de mayor prioridad a otra de menor prioridad. Las direcciones que tiene cada uno de los programas son los tienes:
 - a. Dirección Lógica: Posición de la referencia donde el programa se ubica de forma temporal.
 - b. Dirección relativa: Es la dirección del programa a partir de un punto ya conocido

- c. Dirección real: Es la ubicación física donde el programa se encuentran y los datos que esta posee.
- 3) Protección: Los datos de un programa deben ser protegidas ante interferencias de otros programas, por lo que cada uno tendrá un espacio en memoria que no será llenado a menos que el programa, sin sobrepasar los límites de otro proceso que se esté ejecutando en ese momento, eso evitando que las variables no se vean por la aplicación de un tercero.
- 4) Compartición de recursos: La misma protección debe ser en cierto modo flexible que los recursos puedan ser compartidos, recursos como la tarjeta gráfica o memoria RAM que su capacidad es mucho menor, pero en general los programas utilizan para manejar sus variables, debe tener la capacidad de hacer un uso eficiente y que no sea necesario reiniciar el dispositivo para volver a utilizar los recursos usados con anterioridad.

Llegado a esto, hay 2 grandes maneras de realizar dicho proceso, la primera de ellas por Paginación y la segunda por Segmentación. Según Tanenbaum (2009) vamos a definir las de la siguiente manera:

Técnica de Gestión de Memoria	Paginación	Segmentación
Definición	Técnica de administración de memoria que divide el espacio de direcciones del proceso en bloques de tamaño fijo llamados páginas.	Técnica de administración de memoria que divide el espacio de direcciones del proceso en segmentos de tamaño variable según las necesidades del programa.
Fragmentación	Interna: el tamaño fijo de las páginas puede llevar a desperdicio de espacio dentro de las páginas.	Externa: los segmentos de tamaño variable pueden llevar a espacios libres entre segmentos que no se pueden utilizar eficientemente.
Traducción de Direcciones	Utiliza una tabla de páginas que convierte las direcciones lógicas en direcciones físicas mediante el número de página y el desplazamiento dentro de la página.	Utiliza una tabla de segmentos que convierte las direcciones lógicas en direcciones físicas mediante el número de segmento y el desplazamiento dentro del segmento.

Implementación	Relativamente simple de implementar y eficiente en sistemas con memoria de gran tamaño debido a la uniformidad en el tamaño de las páginas.	Más compleja de implementar debido a la necesidad de gestionar segmentos de tamaño variable y a la posible fragmentación externa.
Uso Común	Ampliamente utilizada en sistemas modernos, incluyendo la mayoría de los sistemas operativos comerciales y de código abierto.	Utilizada en algunos sistemas operativos y aplicaciones específicas donde la gestión detallada de la memoria es crucial.
Ventajas	<ul style="list-style-type: none"> - Elimina la fragmentación externa. - Permite una asignación eficiente de la memoria. - Facilita la protección de memoria entre procesos. 	<ul style="list-style-type: none"> - Los segmentos pueden ser de diferentes tamaños, lo que se ajusta mejor a las necesidades del programa. - Facilita la gestión de datos y código relacionado, ya que cada segmento puede tener un propósito específico (código, datos, pila).
Desventajas	<ul style="list-style-type: none"> - Introduce fragmentación interna debido a la diferencia entre el tamaño de la página y el tamaño de los datos reales. - Complejidad en la gestión de tablas de páginas, especialmente en sistemas con mucha memoria. 	<ul style="list-style-type: none"> - Puede causar fragmentación externa. - Requiere hardware más complejo para la traducción de direcciones. - Dificulta la implementación de técnicas de swapping eficientes debido a la variabilidad en el tamaño de los segmentos.
Videos de explicación	<p>Parte 1: https://www.youtube.com/watch?v=JYyyoeHEAXU&list=PL-TDMQxwQF0HjBg0-dqdOoMxYP9Ob5HP &index=5</p> <p>Parte 2: https://www.youtube.com/watch?v=JYyyoeHEAXU&list=PL-TDMQxwQF0HjBg0-dqdOoMxYP9Ob5HP &index=6</p> <p>Parte 3: https://www.youtube.com/watch?v=JYyyoeHEAXU&list=PL-TDMQxwQF0HjBg0-dqdOoMxYP9Ob5HP</p>	https://www.youtube.com/watch?v=idC9f_srtKA&list=PL-TDMQxwQF0HjBg0-dqdOoMxYP9Ob5HP &index=8

	dqdOoMxYP9Ob5HP &ind ex=7	
Sistemas Operativos	Windows Linux Mac OS	Multics OS/2
Sistemas Operativos Híbridos	Windows en arquitecturas IA-32 Linux en arquitecturas IA-32	

Tabla 1. Paginación y Segmentación de la Memoria

Parte 3. ¿Cómo se administra la memoria en el procesador Intel Core?

Los procesadores Intel Core modernos (Intel Core i3, i5, i7, y i9) utilizan una combinación de paginación y segmentación para la administración de la memoria, aunque la paginación es la técnica predominante.

La segmentación en los procesadores Intel divide la memoria en segmentos lógicos de tamaño variable. Cada segmento posee una base, indicando el inicio y un límite o tamaño del segmento. Los segmentos poseen modos sea de solo lectura, solo escritura o también la particularidad de ser tanto de lectura o escritura. Usualmente en los sistemas operativos modernos la segmentación queda restringida para el modo protegido.

Durante el modo protegido, la segmentación proporciona una capa básica de seguridad y gestión de memoria. Los segmentos se utilizan para definir las áreas de memoria disponibles para el sistema operativo y los programas. La tabla de descriptores de segmento, las cuales son la GDT y LDT contienen los descriptos que definen todos los segmentos durante este modo.

Ya con respecto a lo que es la paginación, los Intel Core cumplen las siguientes funcionalidades:

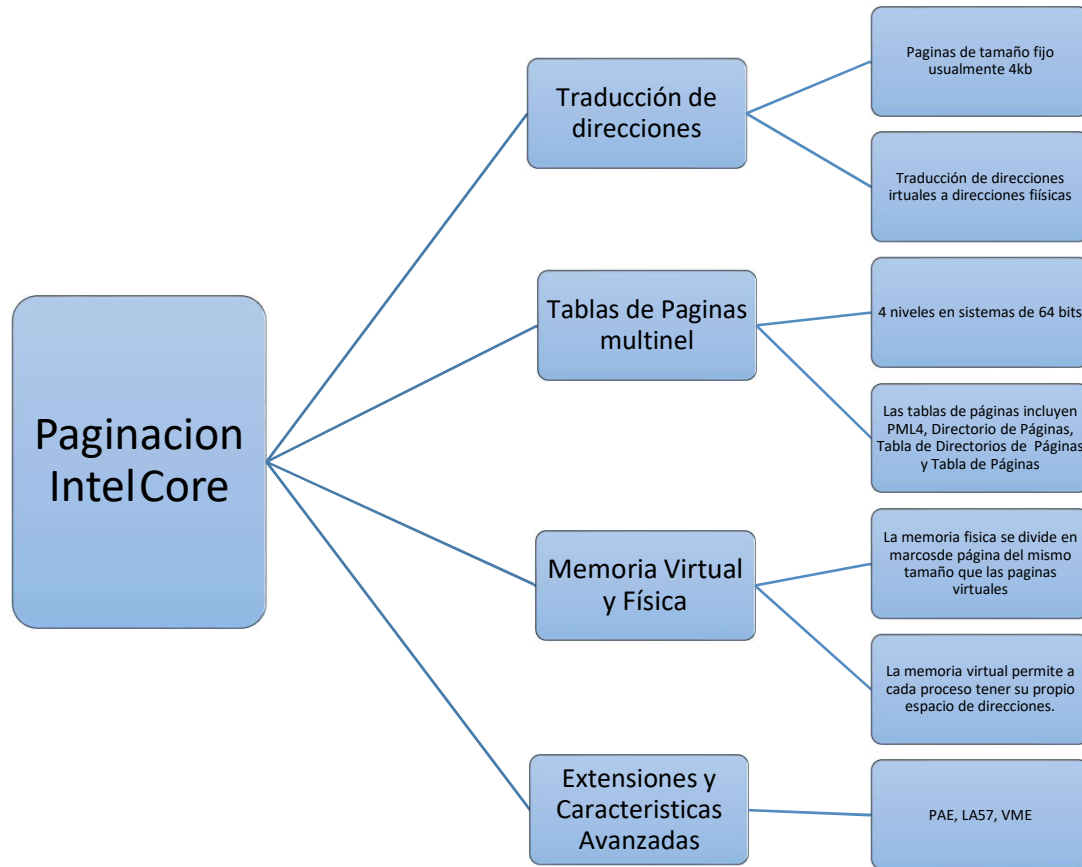


Figura 4. Administración Memoria de un Intel Core

Parte 4. Cuadro comparativo de los algoritmos de reemplazo de páginas ventajas y desventajas.

Algoritmo de reemplazo	Paginas menos recientemente usada (LRU)	Conjunto de Trabajo (Working Set)
Concepto	Reemplaza la página que no ha sido utilizada por el período de tiempo más largo.	Mantiene en memoria las páginas que han sido usadas en un período reciente de tiempo.
Implementación	Utiliza estructuras de datos como pilas, listas enlazadas o tablas hash con conteo.	Requiere mantener un registro del tiempo de referencia de cada página y un reloj deslizante para determinar el conjunto de trabajo actual.
Complejidad	Puede ser costoso en términos de tiempo y espacio debido a la necesidad de actualizar el estado de uso reciente de las páginas continuamente.	costoso en términos de tiempo y espacio porque debe mantener un historial detallado de uso de las páginas y calcular el conjunto de trabajo en tiempo real.
Eficiencia	Funciona bien en la mayoría de los casos, pero puede fallar en escenarios de acceso no secuencial.	Más preciso al manejar patrones de acceso no secuenciales, ya que se adapta al comportamiento del proceso.
Adaptabilidad	No se adapta a cambios en los patrones de acceso de memoria.	Se adapta mejor a cambios en los patrones de acceso, ya que el conjunto de trabajo se recalcula constantemente.

Uso de memoria	Utiliza estructuras de datos adicionales para seguir el orden de uso reciente.	Requiere una cantidad significativa de memoria adicional para almacenar tiempos de referencia y calcular el conjunto de trabajo.
Ventajas	Simple de implementar y razonablemente eficiente en muchos casos prácticos.	Más efectivo en entornos donde los patrones de acceso a la memoria cambian frecuentemente.
Desventajas	No maneja bien los accesos no secuenciales y puede tener sobrecarga de administración de datos.	Complejidad y sobrecarga adicional para mantener y calcular el conjunto de trabajo continuamente.
Casos de uso	Aplicaciones con patrones de acceso predecibles y secuenciales.	Aplicaciones donde los patrones de acceso a la memoria varían significativamente con el tiempo.
Ejemplos de uso	https://www.youtube.com/watch?v=XL3arHkWasY	https://www.youtube.com/watch?v=tcXU8osaEPk

Tabla 2. Cuadro Comparativo de los algoritmos de remplazo LRU y Working Set

Conclusión

Ya llegados a este punto, se darán las respectivas conclusiones del trabajo realizado, el cual se hizo en su mayoría por el tema 3 llamado memorias y se hizo la respectiva lectura del tema 4 Archivos, el cual para esta primera tarea no se vio en gran parte de este último.

Durante la primera parte de la tarea, se observaron las diferencias de como realmente se administra la memoria según el sistema operativo que se tenga instalado, que va a la mano en la mayoría de los casos con el uso que se le dé y la importancia de la memoria para que el uso de ésta sea eficiente y no afecte al resto de dispositivos, donde a modo resumen, en las versiones más actuales de Windows la memoria es parte pero no es el núcleo principal de los procesos, sino más bien un miembro más de la orquesta, que debe trabajar de forma coordinada con el resto de los componentes, mientras en Mainframe la memoria es la integración principal y es donde internamente el sistema operativo realiza las transacciones entre los datos y el direccionamiento de los datos, el cual su tamaño y la velocidad que estos manejan son cruciales para cumplir su labor. En Android, más bien donde su uso, es para dispositivos de menor tamaño, la memoria debe ser capaz de entender, cuales procesos tienen mas prioridad que otros y designarles los recursos cuando el sistema, los requiera. Teniendo que administrar y avisar, cuando se esta alcanzando el limite de las memorias y realizar procesos de limpieza, conocidas como task kill, para que los procesos en primer plano, no se vean afectados.

Ya con respecto a la segunda parte, donde indicamos la diferencia entre paginación y segmentación, se observan las capacidades de cada una de la metodología y la complejidad que conlleva realizar cada una, donde la segmentación es más adecuada para procesos donde la memoria debe ser rápidamente organizada y se le puede sacar el mayor provecho, la segmentada es considerada como la mejor alternativa, pero para utilidades más común donde la memoria es una parte de un complejo sistema, la paginación es más adecuada para este tipo de dispositivos, en la cual su simplicidad hace que sea más rápido de escalar y también conlleva un orden por medio páginas que son referencia por medio de direccionamiento lógicas hacia el espacio real.

Los Intel Core, poseen ambas posibles formas de ordenar su memoria, sin embargo, la segmentación queda en un segundo plano quedando resegada para el modo protegido de los sistemas protegidos, mientras la paginación es la principal manera en cómo se manejan estos dispositivos.

Para finalizar, durante la última parte del documento, se discutieron dos algoritmos para el manejo de la paginación, los cuales funcionan para que el acceso a las direcciones, sean rápidas, cuando existen sea varias repeticiones de la misma página o más bien cuando existen muchos cambios de página. Haciendo que, dependiendo, del uso sea más eficiente una u otra, o alguna otra existente que no se vio durante el documento.

Referencias

Brutti, F. (2023). *Mainframe: qué es, para qué sirve y qué ventajas te ofrece*. Extraído de:

<https://thepower.education/blog/mainframe-que-es-para-que-sirve-y-que-ventajas-te-ofrece>

Equipo editorial, Estece. (2023). *Sistema Operativo*. Argentina Extraído de: <https://concepto.de/sistema-operativo/>

García, M. (2022). *Sistemas de Computación 2 <<Sistemas Operativo>>*. Extraído de:

https://profmatisgarcia.com.ar/uploads/tutoriales/OS3_Memoria.pdf

IBM (2024). *IBM z/OS*. Extraído de: <https://www.ibm.com/es-es/products/zos>

Microsoft (2023). *Administrador de memoria de Windows Kernel-Mode*. Extraído de:

<https://learn.microsoft.com/es-es/windows-hardware/drivers/kernel/windows-kernel-mode-memory-manager>

Statista (2023). *Cuota de mercado mundial de los sistemas operativos para ordenadores de sobremesa de 2010 a 2023*. Extraído de: <https://es.statista.com/estadisticas/634540/sistemas-operativos-para-pc-cuota-de-mercado-mundial/>

Sozpic (2024). *Gestión de memoria en Android*. Extraído de: <https://www.sozpic.com/gestion-de-memoria-en-android/>

Tanembaum, A. (2009). *Sistemas Operativos Modernos*. Tercera edición. Editorial Pearson Educación. México.