

Luis Javier palacio
Universidad Eafit
Colombia
ljpalaciom@eafit.edu.co

Santiago Castrillón Galvis
Universidad Eafit
Colombia
scastrillg@eafit.edu.co

Kevyn Santiago Gómez
Universidad Eafit
Colombia
ksgomezp@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

Sistema de búsqueda y almacenamiento eficiente mediante tabla hash

RESUMEN

Hoy en día existen muchos sistemas de búsqueda y almacenamiento de datos mediante diversas estructuras de datos, sin embargo muchas de ellas no pueden suplir la gran cantidad de información que conlleva el amplio volumen de los archivos, puesto que estamos en la era de la información, por lo que cada vez hay más información que necesitamos almacenar y procesar, esto quiere decir que cada vez se hace más necesario idear sistemas mucho más óptimos y duraderos que cumplan con las necesidades tanto actuales como futuras, evitando así tener que lidiar a futuro con un posible espacio poco eficiente debido a una estructura poco eficiente así como ha sucedido en el pasado, esto se puede evidenciar en estructuras como el ext2, en el cual todavía no se había implementado el journaling, posteriormente la estructura ext3 comenzó a implementarlo, pero una vez más esto no fue suficiente, ya que carecía de extensiones, localización dinámica de inodos, sublocalización de bloques, herramienta de desfragmentación, compresión, y además de esto al no tener una suma de verificación existía el riesgo de que los archivos se corrompieran por fallos del hardware.

Palabras clave

journaling
sublocalización
desfragmentación
inodos
compresión
sistemas de búsqueda y almacenamiento
tabla de hash

Palabras clave de la clasificación de la ACM

Information systems → Data management systems →
Data structures → Data access methods → Data scans

Information systems → Information storage systems →
Record storage systems → Directory structures →
Extent-based file structures

1. INTRODUCCIÓN

En el mundo real se está viendo una masificación de la información de una forma exponencial, la globalización ha permitido que la comunicación a nivel mundial este a disposición de cualquier persona, y de una manera muy sencilla y rápida, por esto la proliferación de la información

es un fenómeno que ha llevado a los sistemas de archivos a estar en constante evolución, ya que a medida que pasa el tiempo las exigencias son mayores, por lo que se hace necesario cambiar a estructuras de datos más eficientes, con mejores herramientas que faciliten y permitan el manejo óptimo de la información. En este proyecto se busca estudiar el proceso por el que han pasado las diversas estructuras de datos que han hecho parte de los sistemas de archivos para analizar los problemas que se han presentado y de esta manera tener una visión más general y completa para buscar una solución que satisfaga las necesidades actuales y tenga en cuenta futuros problemas que se puedan presentar y la forma en la que deben ser afrontados.

2. PROBLEMA

El avance de la tecnología ha provocado que los sistemas de archivos estén en constante evolución, esto significa que el sistema que hace unos años parecía ser el definitivo, ahora no es igual de efectivo, y esto se ha convertido en una tendencia, así que el problema radica en encontrar la forma de optimizar el sistema para que se ajuste a las exigencias de la actualidad y que tenga un funcionamiento a largo plazo, esto tiene una trascendencia de alto impacto pues la tecnología sigue en evolución, pero si no hay una forma de ordenarla entonces habrá un estancamiento en el progreso, lo que puede repercutir en ámbitos sociales y económicos de todo el mundo.

3. TRABAJOS RELACIONADOS

3.1 insertion sort

Estructura de datos para un Sistema de ordenamiento que represente un orden el cual puede ser numeric, alfabetico o alfa-numerico.

El algoritmo insertion sort consta de tomar uno por uno los elementos de un arreglo y recorrerlo hacia su posición con respecto a los anteriormente ordenados. Así empieza con el segundo elemento y lo ordena con respecto al primero. Luego sigue con el tercero y lo coloca en su posición ordenada con respecto a los dos anteriores, así sucesivamente hasta recorrer todas las posiciones del arreglo

3.2 Selection Sort

Estructura de datos para un Sistema de ordenamiento que represente un orden el cual puede ser numerico, alfabetico o alfa-numerico.

El algoritmo selection sort consiste en encontrar el menor de todos los elementos del arreglo e intercambiarlo con el que está en la primera posición. Luego el segundo mas pequeño, y así sucesivamente hasta ordenar todo el arreglo.

3.3 Shellsort

Estructura de datos para un Sistema de ordenamiento que represente un orden el cual puede ser numerico, alfabetico o alfa-numerico.

Denominado así por su desarrollador Donald Shell (1959), ordena una estructura de una manera similar a la del Bubble Sort, sin embargo no ordena elementos adyacentes sino que utiliza una segmentación entre los datos. Esta segmentación puede ser de cualquier tamaño de acuerdo a una secuencia de valores que empiezan con un valor grande (pero menor al tamaño total de la estructura) y van disminuyendo hasta llegar al '1'.

3.4 Bubble Sort

Estructura de datos para un Sistema de ordenamiento que represente un orden el cual puede ser numerico, alfabetico o alfa-numerico

El bubble sort, también conocido como ordenamiento burbuja, funciona de la siguiente manera: Se recorre el arreglo intercambiando los elementos adyacentes que estén desordenados. Se recorre el arreglo tantas veces hasta que ya no haya cambios. Prácticamente lo que hace es tomar el elemento mayor y lo va recorriendo de posición en posición hasta ponerlo en su lugar

4. Tabla de Hash

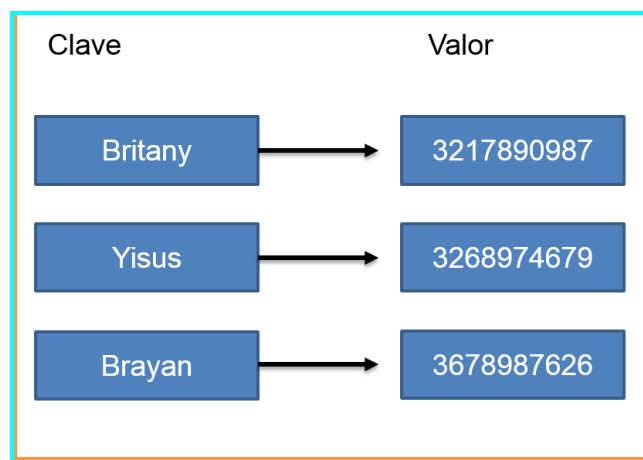


Figura 1: Una tabla de Hash que usa como clave el nombre y como valor se encuentra el numero de telefono.

4.1 Operaciones de la estructura de datos

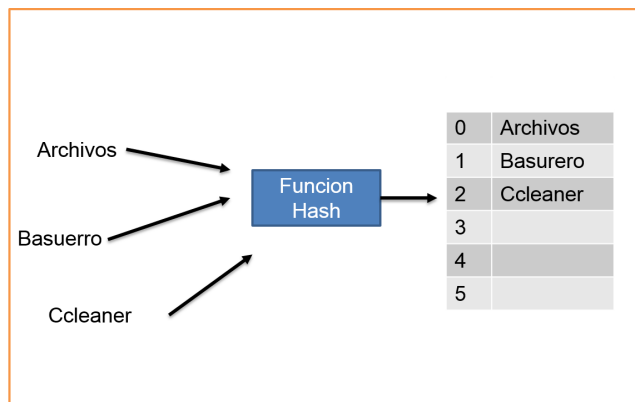
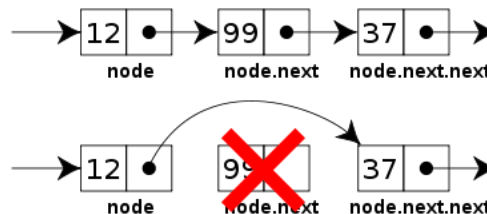
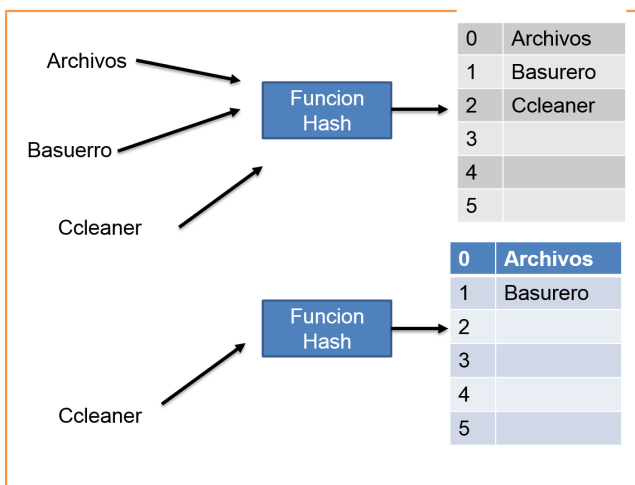


Figura 2: Imagen de una operación de agregación en una tabla de hash.



Gráfica 3: Imagen de una operación de borrado de una tabla de Hash

4.2 Criterios de diseño de la estructura de datos

la estructura de datos es un factor muy importante en el sistema de archivos pues de ella depende el óptimo funcionamiento. Es por esto que hay que tener en cuenta los principales factores que hacen eficiente un sistema de archivos, en este caso específico lo que se necesita es

almacenar una gran cantidad de archivos y encontrarlos de una manera rápida. Para lograr este objetivo se tomaron en cuenta varias estructuras de almacenamiento de datos como lo son linkedList, árbol b, tabla hash, array y estack. Después de analizar las complejidades de estas estructuras en los objetivos específicos de inserción y búsqueda, se llegó a la conclusión de que la estructura más eficaz para esto es la tabla hash, debido a que en la inserción, lo único que se requiere es ingresar una clave, y la función hash se encarga de determinar el lugar idóneo para almacenarla, de tal forma que para búsquedas posteriores, el tiempo de ejecución sea considerablemente más bajo.

4.3 Análisis de Complejidad

| Metodo | Complejidad |
|---------------------|-------------|
| actualizarDirección | $O(n)$ |
| listarContenido | $O(n)$ |
| put | $O(1)$ |
| get | $O(1)$ |
| | |
| | |

Tabla 1: Tabla para reportar la complejidad

4.4 Tiempos de Ejecución

Calculen, (I) el tiempo de ejecución y (II) la memoria usada para las operaciones de la estructura de datos, para el Conjunto de Datos que está en el ZIP

| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Tabla 2: Tiempos de ejecución de las operaciones de la estructura de datos con diferentes conjuntos de datos

4.5 Memoria

Mencionar la memoria que consume el programa para los conjuntos de datos

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |

Tabla 3: Consumo de memoria de la estructura de datos con diferentes conjuntos de datos

4.6 Análisis de los resultados

Expliquen los resultados obtenidos. Hagan una gráfica con los datos obtenidos, como por ejemplo:

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Tabla 4: Análisis de los resultados obtenidos con la implementación de la estructura de datos