ESTRUCTURA DE DATOS PARA LA BÚSQUEDA Y LISTADO DEL CONTENIDO DE UN DIRECTORIO

Santiago Soto
Universidad Eafit
Colombia
ssotom@eafit.edu.co

Andrés Sánchez
Universidad Eafit
Colombia
asanchezc@eafit.edu.co

Jamerson Correa
Universidad Eafit
Colombia
jscorreac@eafit.edu.co

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

RESUMEN
El problema de listar el contenido de un directorio consiste en encontrar eficientemente los archivos y subdirectorios que se encuentran en un directorio. Por ello y por la importancia de la búsqueda de información en los sistemas que manejan grandes cantidades de datos se ha propuesto, como proyecto de clase, encontrar una estructura de datos eficiente para la solución de dicho problema.

Palabras clave

Estructuras de datos → Lectura de archivos → Árbol de búsqueda → Recorrido recursivo

Palabras clave de la clasificación de la ACM

Teoría de la computación → Diseño y análisis de algoritmos → Diseño y análisis de estructuras de datos → Compresión de datos

1. INTRODUCCIÓN

Los sistemas actuales manejan grandes y extensas cantidades de información contenida en archivos. El reto de generar una estructura de datos que distribuya y acomode esta cantidad de archivos se ha convertido en un problema para los desarrolladores de software para lograr la mayor eficiencia en todos sus sistemas.

En el desarrollo de este proyecto se buscará desarrollar un eficiente motor de búsqueda de archivos y subdirectorios de un directorio determinado.

Para la solución de la búsqueda de archivos durante el avance de la tecnología, se diseñó el modelo ext2 y se desarrolló hasta llegar hasta el ext4. El problema con la primera versión de ésta solución no soportaba las cantidades de archivos de los sistemas actuales. Lo que generó su reemplazo por el ext3, pero nuevamente, el avance tecnológico requería utilizar nuevas características no compatibles con el ext3, dando origen al ext4, desarrollado por Google.

2. PROBLEMA

En un sistema donde se requiere manejar grandes cantidades de información almacenada en archivos o subdirectorios, buscar la eficacia a la hora de encontrar uno de estos archivos permitiendo mayores facilidades al usuario, tales como filtros de búsqueda, es el reto que se busca solucionar para optimizar todos los sistemas de información para cualquier cliente, ya sea una compañía que maneje un gran volumen de información o un usuario con un computador personal doméstico.

3. TRABAJOS RELACIONADOS

3.1 Árbol AA

En informática un árbol AA es un tipo de árbol binario de búsqueda auto-balanceable utilizado para almacenar y recuperar información ordenada de manera eficiente. Los árboles AA reciben el nombre de su inventor, Arne Andersson.

Los árboles AA son una variación del árbol rojo-negro, que a su vez es una mejora del árbol binario de búsqueda. A diferencia de los árboles rojo-negro, los nodos rojos en un árbol AA sólo pueden añadirse como un hijo derecho. En otras palabras, ningún nodo rojo puede ser un hijo izquierdo

3.2 Árbol de segmentos

Es una estructura de datos en forma de árbol para guardar intervalos o segmentos. Permite consultar cuál de los segmentos guardados contiene un punto. Este es, en principio, una estructura estática; es decir, su contenido no puede ser modificado una vez que su estructura es construida.

3.3 Árbol AVL

Es un tipo especial de árbol binario ideado por los matemáticos rusos Adelson-Velskii y Landis. Fue el primer árbol de búsqueda binario auto-balanceable que se ideó.

Los árboles AVL están siempre equilibrados de tal modo que, para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha o viceversa.

3.4 Árbol k ario

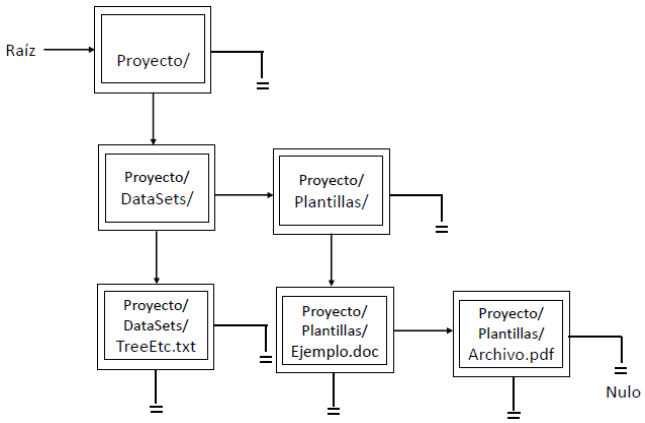
un árbol k-ario es un arraigado árbol en el que cada nodo no tiene más que hijos k. También es conocido a veces como una manera de árbol-k, un árbol N-ario, o un árbol M-ario.

Un árbol k-ario completo es un árbol k-ario donde cada nodo en el mismo nivel 0 tiene hijos k.

4. Estructura tipo árbol general

Implementación personal de una estructura tipo árbol conformada por un conjunto de datos de tipo nodo que representará cada directorio o fichero dentro de la estructura, donde cada nodo tendrá relaciones con otros nodos, nodo padre, que será el directorio contenedor del nodo actual; el nodo hijo, que será un subdirectorio o un fichero del nodo actual; y un nodo hermano, que son directorios o ficheros dentro del mismo directorio. En esta estructura el único nodo sin nodo padre o hermano será el nodo raíz.

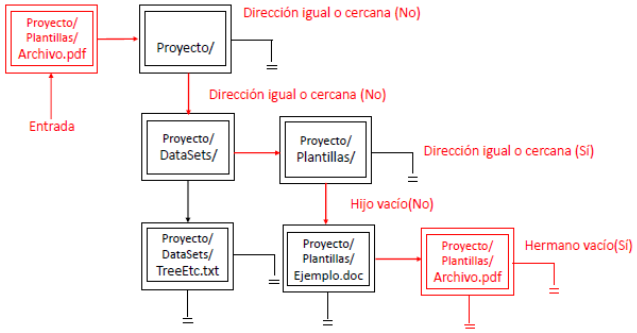
Diseño de la estructura de datos:



4.1 Operaciones de la estructura de datos

4.1.1 addNodo: añadir fichero o directorio a la estructura.

Esta operación verifica si un nodo, en este caso iniciando desde el nodo padre, posee o no hijos, en caso de no poseerlos éste se encarga de añadirlos haciendo a sí mismo un llamado recursivo.



4.2 Criterios de la estructura

La propiedad de las estructuras tipo árbol de jerarquizar los datos, hacen de la estructura utilizada la apropiada para generar la correcta asignación de subdirectorios y ficheros a cada directorio, por otro lado, se asemeja a la estructura de la distribución dada por un sistema operativo de cada dato dentro del disco duro. Además, con el llamado recursivo para las funciones de nuestra estructura, se reducen notablemente la complejidad y por ende el tiempo de ejecución de cada funcionalidad de la estructura.

4.3 Análisis de las complejidades

Función	Complejidad
addNodo	O(n)

4.4 Tiempo de ejecución

Funcion	conjunto ejemplito.txt	conjunto treeEtc.txt	conjunto juegos.txt
Creación	0,3 ms	33,19 ms	

Tabla 1. Tiempos de ejecución de las operaciones de la estructura de datos con diferentes conjuntos de datos

4.5 Memoria usada

Estructura	conjunto ejemplito.txt	conjunto treeEtc.txt	conjunto juegos.txt
Memoria	4.458.728 Mb	29.768.304MB	

4.6 Análisis de los resultados

Estructura	ejemplito	juegos	treeEtc
Espacio en el heap	4.458.728 Mb	29.768.304 MB	
tiempo de creación	0,3 ms	33,19 ms	
tiempo de búsqueda (“a”)			
tiempo de búsqueda (“zyzzvya”)			
tiempo de búsqueda (“aerobacteriolocally”)			
Tiempo búsqueda todas las palabras			

REFERENCIAS

1. Wikiwand, Árbol AA. http://www.wikiwand.com/es/%C3%81rbol_AA
2. Wikipedia, Árbol de segmento. https://es.wikipedia.org/wiki/%C3%81rbol_de_segmento

3. Wikipedia, Árbol AVL.
https://es.wikipedia.org/wiki/%C3%81rbol_AVL
4. Wikipedia, Árbol k-
ario.https://es.wikipedia.org/wiki/%C3%81rbol_k-ario