



Centro Universitario de Ciencias Exactas e Ingenierías.

Departamento para la integración ciber humana.

Sistemas operativos.

Becerra Velázquez Violeta Rocío.

SAUL EMMANUEL YAÑEZ SALAZAR
220656484.

Ingeniería en computación.

D04

Entrada y salida.

Domingo 27 de abril del 2025.

(vacías)					
 Quiz complementario actividad 13	2.70 %	4.00	0-4	100.00 %	2.70 %

Acciones posibles en el manejo de archivos y directorios.

Acciones en directorios:

Crear directorio

Establece una nueva carpeta dentro de una ubicación especificada, proporcionando un espacio organizado para almacenar archivos relacionados.

Eliminar directorio

Suprime un directorio vacío del sistema de archivos, liberando el espacio que ocupaba.

Listar contenido

Muestra los archivos y subdirectorios contenidos en un directorio, facilitando la navegación y organización.

Cambiar nombre

Modifica el nombre de un directorio, permitiendo una identificación más clara o acorde con su contenido.

Mover directorio

Traslada un directorio de una ubicación a otra dentro del sistema de archivos, reestructurando la jerarquía según sea necesario.

Copiar directorio

Duplica un directorio junto con su contenido en una nueva ubicación, preservando la estructura original.

Acceder a directorio

Cambia el directorio de trabajo actual a otro especificado, permitiendo operar en diferentes ubicaciones del sistema de archivos

Acciones en archivos.

Crear archivo

Genera un nuevo archivo vacío o con contenido inicial en una ubicación determinada.

Eliminar archivo

Suprime un archivo del sistema de archivos, liberando el espacio que ocupaba.

Renombrar archivo

Cambia el nombre de un archivo, facilitando su identificación o alineación con convenciones de nomenclatura.

Mover archivo

Traslada un archivo de una ubicación a otra, reorganizando el sistema de archivos según sea necesario.

Copiar archivo

Duplica un archivo en una nueva ubicación, permitiendo su respaldo o distribución sin alterar el original.

Leer archivo

Accede al contenido de un archivo para su visualización o procesamiento.

Escribir en archivo

Modifica o agrega contenido a un archivo existente, actualizando su información.

Ejecutar archivo

Lanza la ejecución de un archivo, generalmente un script o programa, para realizar tareas específicas.

Establecer permisos

Define los derechos de acceso de usuarios y grupos sobre un archivo, controlando su lectura, escritura y ejecución.

Verificar existencia

Comprueba si un archivo o directorio existe en una ubicación específica, útil para validaciones en scripts y programas.

Indices.

En el ámbito de la informática, el término **índice** se refiere a una estructura de datos que facilita la localización eficiente de información dentro de un conjunto más amplio, como una base de datos, un sistema de archivos o una estructura de programación. Su función principal es optimizar el acceso a los datos, permitiendo búsquedas rápidas y eficientes.

Definición general.

Un índice es una lista organizada que asocia claves (como palabras clave o identificadores) con ubicaciones específicas de datos. Esta organización permite acceder directamente a los datos sin necesidad de recorrer todo el conjunto, similar a cómo un índice en un libro señala la página exacta donde se encuentra un tema específico.

Tipos.

Índice en Bases de Datos

En bases de datos, un índice es una estructura que mejora la velocidad de las operaciones de búsqueda y recuperación de datos. Funciona como una tabla que almacena claves de búsqueda junto con referencias a las ubicaciones de los datos correspondientes. Los índices pueden ser primarios, secundarios o de clúster, dependiendo de cómo se relacionan con la clave primaria de la tabla y cómo se almacenan los datos. (Authors, 2022)

2. Índice en Sistemas de Archivos

En sistemas de archivos, un índice puede referirse a estructuras como los i-nodos en sistemas Unix, que almacenan información sobre archivos y directorios, como permisos, propietario y ubicación de los datos. Estos índices permiten al sistema operativo localizar y acceder rápidamente a los archivos almacenados en el disco. (Freitas, s.f.)

3. Índice en Programación

En programación, un índice es una variable que indica la posición de un elemento dentro de una estructura de datos, como un arreglo o lista. Por ejemplo, en un arreglo de 10 elementos, los índices van del 0 al 9, permitiendo acceder a cada elemento de manera eficiente.

Dispersiones.

En informática, una función de dispersión (o hash function) es un algoritmo que toma una entrada (o clave) y la mapea a un valor de longitud fija, generalmente un número entero. Este valor se utiliza como índice para almacenar o recuperar datos en una estructura de datos como una tabla hash.

La calidad de una función de dispersión es crucial para el rendimiento de la tabla hash. Una función bien diseñada distribuye las claves de manera uniforme a través de las ubicaciones disponibles, minimizando las colisiones y asegurando tiempos de acceso eficientes. (Sullivan, s.f.)

Metodos de dispersiones:

En el ámbito de la informática, particularmente en estructuras de datos como las tablas hash, la dispersión se refiere al proceso de asignar claves a ubicaciones específicas dentro

de una tabla para facilitar el acceso eficiente a los datos. Existen diversos métodos para lograr una dispersión efectiva, cada uno con sus características y aplicaciones particulares.

Método de División: este es uno de los métodos más sencillos y rápidos para generar un valor hash. Consiste en dividir la clave entre un número primo y tomar el residuo como valor hash. Se utiliza ampliamente debido a su simplicidad y eficiencia.

Método del Cuadrado Medio: en este método se eleva al cuadrado la clave, se extrae una parte del resultado (normalmente los dígitos del medio) y se utiliza como valor hash. Se aplica cuando se busca una mejor distribución de los valores.

Método de Plegado: la clave se divide en varias partes, estas se suman, y el resultado se toma como valor hash. Es útil para claves alfanuméricas largas o con patrones estructurados.

Método de Multiplicación: este método consiste en multiplicar la clave por una constante fraccionaria entre 0 y 1, tomar la parte fraccionaria del resultado, multiplicarla por el tamaño de la tabla, y tomar la parte entera como valor hash. Proporciona una dispersión uniforme si se elige bien la constante.

Existen además técnicas para resolver colisiones, que son situaciones en las que dos claves diferentes generan el mismo valor hash.

Encadenamiento separado: cada posición de la tabla contiene una lista donde se almacenan todas las claves que colisionan en esa posición. Es una técnica efectiva cuando el número de colisiones puede ser alto.

Direccionamiento abierto: cuando ocurre una colisión, se busca otra posición vacía dentro de la tabla. Existen variantes como sondeo lineal (se busca secuencialmente), sondeo cuadrático (se aumenta el espacio entre intentos de forma cuadrática), y doble dispersión (se usa una segunda función hash para calcular la nueva posición).

Soluciones de colisiones.

Encadenamiento Separado (Separate Chaining)

Esta técnica consiste en que cada posición de la tabla hash contiene una lista (o "bucket") que almacena todas las claves que generan el mismo valor hash. Cuando ocurre una colisión, la nueva clave se añade al final de la lista correspondiente. Este método es sencillo de implementar y permite una gestión eficiente de las colisiones, aunque puede requerir más memoria debido al almacenamiento de listas adicionales.

Dirección Abierta (Open Addressing)

En este enfoque, todas las claves se almacenan directamente en la tabla hash. Cuando ocurre una colisión, se busca otra posición disponible en la tabla según una secuencia de sondeo. Existen varios métodos de sondeo:

- Sondeo Lineal (Linear Probing): Se verifica secuencialmente la siguiente posición disponible.
- Sondeo Cuadrático (Quadratic Probing): La distancia entre intentos aumenta cuadráticamente.
- Doble Hashing (Double Hashing): Se utiliza una segunda función hash para determinar el intervalo de sondeo.

La dirección abierta es eficiente en términos de memoria, ya que no requiere estructuras adicionales, pero puede ser más susceptible a problemas de clustering.

Hashing Cuckoo (Cuckoo Hashing)

Este método utiliza dos o más funciones hash y mantiene múltiples tablas. Cuando una clave colisiona en una posición, se desplaza la clave existente a su posición alternativa, y así sucesivamente, hasta encontrar una posición vacía. Este enfoque garantiza tiempos de búsqueda constantes en el peor de los casos, pero puede requerir reestructuraciones complejas en caso de ciclos de desplazamiento. (Wikipedia Authors, 26)

Hashing Perfecto (Perfect Hashing)

Se diseña una función hash específica para un conjunto conocido de claves, de modo que no ocurran colisiones. Es ideal para conjuntos estáticos donde las claves no cambian, ya que permite búsquedas en tiempo constante sin colisiones. Sin embargo, su implementación puede ser compleja y no es adecuada para conjuntos dinámicos.

Hashing Coalescente (Coalesced Hashing)

Combina aspectos del encadenamiento separado y la dirección abierta. Cuando ocurre una colisión, la nueva clave se almacena en una posición libre determinada por una secuencia de sondeo, y se enlaza con la clave original mediante punteros. Este método busca reducir el espacio adicional requerido por el encadenamiento separado y minimizar el clustering de la dirección abierta.

Conclusiones.

En esta tarea pudimos recabar como una parte tan esencial y tan presente en la informática se encuentra en uso todos los días, los índices son pequeños espacios que permiten encontrar cualquiera que sea el elemento usando “coordenadas” si así lo quisieramos llamar, ya que actúan de la misma manera que lo hace un índice de un libro, te dice el contenido principal del elemento, y tu solo te encargas de buscar lo que realmente necesitas, en lugar de buscar por todo el libro.

Bibliografía

Authors, W. (29 de Noviembre de 2022). *Índices (bases de datos)* . Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Índice_%28base_de_datos%29?utm_source=chatgpt.com

Freitas, G. (s.f.). *Sistemas de archivos*. Obtenido de StudySmarter:
https://www.studysmarter.es/resumenes/ciencias-de-la-computacion/sistemas-informaticos/sistemas-de-archivos/?utm_source=chatgpt.com

Sullivan, D. G. (s.f.). *Harvard Files*. Obtenido de Harvard:
https://cscis111.sites.fas.harvard.edu/files/lectures/unit9-4.pdf?utm_source=chatgpt.com

Wikipedia Authors. (2025 de Abril de 26). *Cuckoo Hashing*. Obtenido de Wikipedia:
https://en.wikipedia.org/wiki/Cuckoo_hashing?utm_source=chatgpt.com