



SISTEMAS OPERATIVOS

ACTIVIDAD DE APRENDIZAJE 13

TAREA 7

Profesor:

VIOLETA DEL ROCIO BECERRA VELAZQUEZ

VERDUZCO ROSALES LUIS ENRIQUE

223992388

Ingeniería en Computación

16/11/25

CUCEI DIVTIC D04

ÍNDICE

1. Quiz.....	2
2. Acciones posibles en el manejo de Archivos y Directorios.....	2
3. Resumen de Temas de Estructuras de Datos.....	3
2. Direccionamiento Abierto (Open Addressing).....	5
Conclusión.....	5
Bibliografía.....	5

Actividad de Aprendizaje 11

1. Quiz

   Luis Enrique Verduzco Rosales 

Entradas-salidas

Instrucciones.

- Una vez que se vean los algoritmos de planificación de disco en clase, realizar el presente quiz, para ello lea con atención y conteste según corresponda.
- Realizar una captura de pantalla para anexarlo a la entrega de la actividad 13.

Intentos permitidos: 6

Este examen se abrió en lunes, 20 de octubre de 2025, 07:00

Este examen se cerrará en miércoles, 5 de noviembre de 2025, 23:00

Método de calificación: Calificación más alta

Resumen de sus intentos previos

Intento	Estado	Calificación / 3.00	Calificación / 4.00
1	Terminado Enviado miércoles, 5 de noviembre de 2025, 20:01	3.00	4.00

Calificación más alta: 4.00 / 4.00.

Reintentar el examen

2. Acciones posibles en el manejo de Archivos y Directorios

A continuación, se listan las operaciones fundamentales que los sistemas operativos y las aplicaciones realizan para gestionar archivos (datos) y directorios (carpetas que organizan los archivos).

- Crear (Create):
 - Descripción: Esta acción genera un nuevo archivo (generalmente vacío) o un nuevo directorio en una ubicación específica del sistema de archivos. Requiere un nombre y una ruta.
- Leer (Read):
 - Descripción: Permite acceder al contenido de un archivo. La aplicación abre el archivo y extrae sus datos (parcial o totalmente) para procesarlos, mostrarlos o copiarlos en la memoria.
- Escribir/Modificar (Write/Modify):
 - Descripción: Altera el contenido de un archivo existente. Esto puede implicar sobrescribir los datos actuales o añadir (append) nuevos datos al final del archivo.
- Eliminar (Delete):
 - Descripción: Remueve permanentemente un archivo o un directorio del sistema de archivos. Si se elimina un directorio, generalmente también se deben eliminar todos los archivos y subdirectorios que contiene.

- **Renombrar (Rename):**
 - Descripción: Cambia el nombre de un archivo o directorio existente sin alterar su contenido o, en la mayoría de los sistemas, su ubicación.
- **Mover (Move):**
 - Descripción: Cambia la ubicación física de un archivo o directorio, trasladándolo de un directorio (ruta) a otro. En el mismo volumen de almacenamiento, esto suele ser una operación rápida que solo actualiza los punteros del sistema de archivos.
- **Copiar (Copy):**
 - Descripción: Crea un duplicado exacto de un archivo o directorio en una nueva ubicación o con un nuevo nombre. A diferencia de "mover", el original permanece en su lugar.
- **Listar (List):**
 - Descripción: Muestra el contenido de un directorio, enumerando los archivos y subdirectorios que se encuentran dentro de él. A menudo incluye metadatos como el tamaño o la fecha de modificación.
- **Obtener/Modificar Propiedades (Get/Set Properties):**
 - Descripción: Permite consultar o cambiar los metadatos de un archivo o directorio. Esto incluye:
 - Permisos: Controlar quién puede leer, escribir o ejecutar el archivo (ej. chmod en Linux).
 - Atributos: Banderas especiales como "solo lectura", "oculto" o "archivo del sistema".
 - Metadatos: Consultar el tamaño, las fechas de creación/modificación/acceso, y el propietario.

3. Resumen de Temas de Estructuras de Datos

- **Índices**

Un índice es una estructura de datos auxiliar diseñada para acelerar la recuperación de información en un conjunto de datos más grande (como una tabla en una base de datos o un archivo).

Su función es análoga al índice de un libro: en lugar de leer el libro entero (escanear la tabla completa) para encontrar un tema, se consulta el índice (una estructura pequeña y ordenada), que indica la "página" (la ubicación en el disco o la fila) donde se encuentra el dato exacto.

Los índices se crean sobre una o más columnas (las "claves" del índice). Si bien aceleran drásticamente las operaciones de búsqueda (consultas SELECT), tienen un costo: ralentizan las operaciones de escritura (como INSERT, UPDATE, DELETE), ya que el índice también debe ser actualizado cada vez que los datos cambian.

- **Dispersión (Hashing)**

La dispersión (o hashing) es una técnica que utiliza una función hash para mapear datos de un tamaño arbitrario (como un nombre de usuario, un objeto complejo, etc.) a un valor de tamaño

fijo, generalmente un número entero. Este entero se utiliza como un "índice" o "dirección" en una estructura de datos llamada tabla hash (o tabla de dispersión).

El objetivo principal del hashing es permitir un acceso, inserción y eliminación de datos en tiempo promedio casi constante, $O(1)$, lo cual es extremadamente rápido.

Una buena función hash debe ser:

1. Rápida de calcular.
2. Determinista: La misma entrada siempre debe producir la misma salida (el mismo hash).
3. Distribuida uniformemente: Debe "dispersar" las entradas de manera uniforme por todas las posibles salidas para minimizar las colisiones.

Métodos de Dispersión

Los "métodos de dispersión" se refieren a las diferentes funciones hash o algoritmos matemáticos utilizados para calcular el índice (la dirección) a partir de la clave. El objetivo es tomar la clave (que puede ser un número grande o un texto) y convertirla en un índice válido dentro del tamaño de la tabla.

Algunos métodos comunes incluyen:

- Método de la División (o Aritmética Modular):
 - Fórmula: $h(k) = k \bmod M$
 - Descripción: Se toma la clave k y se calcula el residuo (módulo) de la división por M , donde M es el tamaño de la tabla. Es el método más simple y rápido. Funciona mejor si M es un número primo.
- Método de la Multiplicación:
 - Descripción: Se multiplica la clave k por una constante A (un número entre 0 y 1). Se toma la parte fraccionaria de ese resultado, se multiplica por el tamaño de la tabla M , y se toma la parte entera. Es menos sensible a la elección de M .
- Mitad del Cuadrado (Mid-Square):
 - Descripción: Se eleva la clave al cuadrado y se extraen los dígitos (o bits) del centro del resultado. La idea es que los dígitos del medio suelen ser más "aleatorios".

Solución de Colisiones

Una colisión ocurre cuando la función hash genera el mismo índice (dirección) para dos o más claves diferentes. Dado que no podemos almacenar dos datos distintos en la misma ranura exacta, necesitamos una estrategia para manejar esto.

Existen dos enfoques principales para solucionar colisiones:

1. Encadenamiento (Chaining) o Direccionamiento Separado

- Descripción: En este método, cada índice de la tabla hash no almacena el dato directamente, sino que apunta al inicio de una estructura de datos secundaria, comúnmente una lista enlazada (o a veces un árbol).

- Funcionamiento: Si las claves "Juan Pérez" y "Ana Gómez" generan el mismo hash (ej. índice 5), ambas se añaden a la lista enlazada que "cuelga" del índice 5. Para buscar, se calcula el hash, se va a esa lista y se recorre la lista hasta encontrar el elemento.

2. Direccionamiento Abierto (Open Addressing)

- Descripción: En este método, todos los elementos se almacenan dentro de la misma tabla (el array principal). Cuando ocurre una colisión (la ranura de destino ya está ocupada), el algoritmo "sondea" (busca) la siguiente ranura libre en la tabla según una regla predefinida.
- Sub-métodos (tipos de sondeo):
 - Sondeo Lineal (Linear Probing): Si la ranura i está ocupada, prueba con $i+1$, $i+2$, $i+3$, ... (Problema: Tiende a crear "grupos" o clusters).
 - Sondeo Cuadrático (Quadratic Probing): Si la ranura i está ocupada, prueba con $i+1^2$, $i+2^2$, $i+3^2$, ... (Ayuda a dispersar más los elementos y evitar el clustering primario).
 - Doble Dispersión (Double Hashing): Se utiliza una segunda función hash para determinar el "tamaño del salto" en el sondeo. (Es el método de direccionamiento abierto más eficiente).

Conclusión

En conclusión, aprendí que en conjunto, todos estos temas describen cómo los sistemas de software organizan y encuentran datos de manera eficiente.

Los archivos y directorios son la forma en que vemos la información organizada.

Los índices y la dispersión (hashing) son las técnicas internas e invisibles que usan los sistemas operativos y las bases de datos para hacer que esa organización sea increíblemente rápida, permitiendo encontrar un archivo o un dato específico al instante, sin tener que buscar en todo el disco.

Bibliografía

- Universidad Nacional del Sur. (s.f.). *07-FileSystemsExtra2_BW* [Archivo PDF]. cs.uns.edu.ar. https://cs.uns.edu.ar/~gd/soyd/clases/07-FileSystemsExtra2_BW.pdf
- Daniel Santana. (s.f.). *Manejo de indices*. Blogspot. <https://dan1456bd.blogspot.com/p/manejo-de-indices.html>
- César Vaca. (s.f.). *Tema 5* [Archivo PDF]. Universidad de Valladolid. <https://www.infor.uva.es/~cvaca/asigs/doceda/tema5.pdf>
- Franco Martínez, E. A. (s.f.). *Tema 05: "Tablas Hash"* [Archivo PDF]. <https://docencia.eafranco.com/materiales/estructurasdedatos/05/Tema05.pdf>

Actividad de Aprendizaje 11

- Bruno López. (s.f.). *Colisiones* [Archivo PDF]. Tecnológico Nacional de México, Campus Nuevo Laredo.
https://nlaredo.tecnm.mx/takeyas/Apuntes/Administracion_Archivos/Apuntes/Colisiones.PDF
- Educative.io. (s.f.). *Hash table collision resolution*.
<https://www.educative.io/answers/hash-table-collision-resolution>
- MLG. (s.f.). *TP-5* [Archivo PDF]. Universidad Nacional del Sur.
<https://cs.uns.edu.ar/~mlg/ed/downloads/PR%C3%81CTICOS/TP-5.pdf>
- GeeksforGeeks. (s.f.). *Open addressing collision handling technique in hashing*.
<https://www.geeksforgeeks.org/dsa/open-addressing-collision-handling-technique-in-hashing/>