Trabajo Práctico de Investigación Archivos

Alumno: Leandro Panozzo

Año: 2023

Materia: Laboratorio de Programacion 1

Profesora: Carolina Vargas



**Ejercicio N°1**: Definir los siguientes conceptos

• Archivo

• Mezcla de Archivos

• Apareo de Archivos

• Corte de Control

**Ejercicio N°2:** Describir las siguientes operaciones

• Mezcla de Archivos

• Apareo de Archivos

• Corte de Control

**Ejercicio N°3:**

• ¿Qué diferencia hay entre un arreglo y un archivo?

• ¿Qué diferencia hay entre mezcla y apareo?

• ¿Cómo se clasifican los archivos según el método de acceso?

• ¿Cómo se clasifican los archivos según el tipo de contenido?

• ¿Qué diferencia hay entre un archivo de texto y un archivo binario?

• ¿Por qué no se puede abrir un archivo?

• ¿Por qué es necesario cerrar los archivos al final del proceso?

**Ejercicio N°3**: Indique en C las instrucciones para realizar las siguientes operaciones

**1. ¿Qué significa este tipo de apertura?**

r: ……………………………………………………………………..

w: …………………………………………………………………..

a: …………………………………………………………………

r+:…………………………………………………………………

w+: ………………………………………………………………..

a+: …………………………………………………………………..

rb: …………………………………………………………………..

wb: ……………………………………………………………….

ab: ……………………………………………………………….

r+b: …………………………………………………………………………..

w+b: ……………………………………………………………………………..

a+b: ……………………………………………………………………………..

**2. ¿Qué significa lo siguiente?**

a. t: modo texto. Normalmente es el modo por defecto. Se suele omitir.

b. b: modo binario.

**3. Indicar para que se usan las siguientes funciones**

|  |  |
| --- | --- |
| Función | Significado |
| fopen() |  |
| fclose() |  |
| fgets() |  |
| fputs() |  |
| fseek() |  |
| fprintf() |  |
| fscanf() |  |
| feof() |  |
| ferror() |  |
| rewind() |  |
| remove() |  |
| fflush() |  |

**4. ¿Qué significa “FILE \*nombrearchivo”?**

**5. ¿Qué significa el siguiente trozo de programa? ¿Qué tipo de archivo se está abriendo?** FlLE \*fich; if ((fich = fopen("nomfich.dat", "r")) == NULL) printf ( " Error en la apertura. Es posible que el fichero no exista \n ");

**6. ¿Qué tipo de apertura tienen los siguientes archivos?**

FILE \* datosdatos = fopen (“nombres.dat”,”r”); datos = fopen (“nombres.dat”, “w”); datos = fopen (“nombres.dat”, “a”); datos = fopen (“nombres.dat”, “ra”);

**7. ¿Qué representa el siguiente trozo de programa?**

FILE \*parch; if((parch=fopen(“c:\\banco.dat”,”rb”))==NULL)//Se abre en modo lectura printf(“\nEl archivo no puede ser abierto”); if((fclose(parch))= = -1) //Se cierra el archivo printf(“\nNo se pudo cerrar el archivo”); else printf(“\nEl archivo se cerro exitosamente”);

1) **Archivo**: Es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son de igual tipo y constan a su vez de diferentes entidades de nivel más bajos denominadas campos.

**Mezcla de Archivos**: La mezcla de archivos es un proceso en el cual se combinan o fusionan dos o más archivos de manera ordenada, creando un nuevo archivo resultante. Se utiliza comúnmente en procesos de ordenamiento externo o cuando se necesita fusionar datos de diferentes fuentes en un único archivo. La mezcla de archivos implica la comparación y combinación de registros de los archivos de origen de acuerdo a un criterio específico, como un valor clave.

**Apareo de Archivos:** El apareo de archivos, también conocido como "merge-sort", es un algoritmo utilizado para combinar y ordenar dos o más archivos previamente ordenados en un solo archivo ordenado. Este proceso se basa en la comparación de los registros de los archivos de origen, utilizando un criterio de ordenación, generalmente basado en un valor clave. El apareo de archivos es ampliamente utilizado en algoritmos de ordenamiento externo, donde los datos no caben completamente en la memoria principal.

**Corte de Control:** El corte de control es una técnica utilizada en procesos de clasificación y agrupación de datos en archivos. Consiste en identificar registros que comparten un atributo común, llamado campo de control, y separarlos en grupos o subarchivos. El corte de control permite organizar y resumir grandes volúmenes de datos en forma estructurada y facilita su posterior análisis o procesamiento. Por lo general, se utiliza en operaciones de totalización, cálculo de subtotales o generación de informes estadísticos.

2)

**Mezcla de Archivos:** La mezcla de archivos es una operación en la cual se combinan o fusionan dos o más archivos en uno solo, manteniendo un orden específico de los registros. Este proceso implica la comparación de los registros de los archivos de origen y su inserción ordenada en el archivo resultante. Se utiliza comúnmente en procesos de ordenamiento externo o cuando se necesita combinar datos de diferentes fuentes en un solo archivo.

**Apareo de Archivos:** El apareo de archivos es una operación que consiste en combinar y ordenar dos o más archivos previamente ordenados en un solo archivo ordenado. Este proceso se basa en la comparación de los registros de los archivos de origen utilizando un criterio de ordenación, como un valor clave. Los registros se toman uno a uno de cada archivo y se insertan en orden en el archivo resultante. El apareo de archivos es utilizado en algoritmos de ordenamiento externo, donde los datos no caben completamente en la memoria principal.

**Corte de Control:** El corte de control es una operación que implica la separación o agrupación de registros en un archivo según un atributo común, llamado campo de control. Los registros que comparten el mismo valor en el campo de control se agrupan juntos en un subarchivo o se generan resúmenes estadísticos para cada grupo. Esta operación es útil para realizar análisis y cálculos sobre subconjuntos de datos relacionados. Por ejemplo, en un archivo de ventas, se puede realizar un corte de control por fecha para obtener las ventas totales de cada día.

**3)  
Diferencia entre un arreglo y un archivo:**

Un arreglo es una estructura de datos en la memoria que almacena elementos del mismo tipo de forma contigua. Los arreglos son utilizados para almacenar y acceder a conjuntos de datos en la memoria principal de un programa. Pueden ser modificados y accedidos de forma eficiente en tiempo constante, pero su tamaño está limitado por la memoria disponible.

Por otro lado, un archivo es una unidad de almacenamiento persistente en un sistema de archivos. Los archivos se utilizan para guardar información de forma permanente en un dispositivo de almacenamiento, como un disco duro. Los archivos pueden contener cualquier tipo de datos y pueden ser de tamaño mucho mayor que la memoria principal. Acceder a los datos de un archivo implica operaciones de lectura y escritura en el disco, lo que generalmente es más lento que acceder a la memoria principal.

**Diferencia entre mezcla y apareo:**

La mezcla de archivos y el apareo de archivos son conceptos relacionados, pero con algunas diferencias:

* **Mezcla de archivos:** Es un proceso en el cual se combinan dos o más archivos para crear un nuevo archivo resultante, sin tener en cuenta el orden de los registros. La mezcla de archivos puede utilizarse para combinar información de diferentes fuentes o para unir archivos sin ninguna consideración de orden.
* **Apareo de archivos:** Es un proceso en el cual se combinan y ordenan dos o más archivos previamente ordenados en un solo archivo ordenado. El apareo de archivos implica la comparación de los registros de los archivos de origen y su inserción en el archivo resultante en el orden correcto, basado en un criterio de ordenación (por ejemplo, un valor clave). Este proceso se utiliza comúnmente en algoritmos de ordenamiento externo.

**Clasificación de archivos según el método de acceso:**

Depende del método físico en el que se organiza la información

**Archivos secuenciales:** Los registros se leen o escriben secuencialmente, uno después del otro, en el mismo ordenen el que se situaron, para llegar al registro deseado se recorren todos los precedentes

**Archivos de acceso directo:** Permite el acceso directo a registros utilizando una clave o identificador único asociado a cada registro. Los registros se pueden buscar y recuperar directamente según el valor de la clave, sin necesidad de recorrer el archivo secuencialmente.

**Clasificación de archivos según el tipo de contenido:**

**Archivos de texto:** Almacenan información legible por humanos en forma de texto. Los caracteres se codifican utilizando una codificación de caracteres (como ASCII o UTF-8). Los archivos de texto pueden contener líneas de texto, separadas por caracteres de nueva línea.

**Archivos binarios**: Almacenan información en una forma más compacta y directa para su procesamiento por parte de una computadora. Los archivos binarios pueden contener datos estructurados y no estructurados, como números, imágenes, sonidos, programas ejecutables, entre otros. Los datos se representan en una forma más cruda y eficiente en términos de espacio de almacenamiento.

**Diferencia entre un archivo de texto y un archivo binario:**

Un archivo de texto almacena información en formato de texto legible por humanos. Los caracteres se codifican utilizando una codificación de caracteres específica, y los datos se almacenan en líneas de texto separadas por caracteres de nueva línea. Estos archivos se pueden editar y leer utilizando un editor de texto.

Los archivos de texto se caracterizan por ser planos, es decir, todas las letras tienen el mismo formato y no hay palabras subrayadas, en negrita, o letras de distinto tamaño o ancho.

Un archivo binario almacena información en una forma más compacta y directa para su procesamiento por parte de una computadora. Los datos en un archivo binario se representan en su forma cruda y se pueden utilizar para almacenar cualquier tipo de información, como números, imágenes, sonidos, programas ejecutables, entre otros. Estos archivos no son fácilmente legibles por humanos y generalmente requieren programas o algoritmos específicos para leer y procesar su contenido.

En un archivo binario se guardan datos con distinto formato, es decir se pueden guardar caracteres mezclados con enteros y flotantes.

Por qué es necesario cerrar los archivos al final del proceso:  
Un error en el cierre de una secuencia puede generar todo tipo de problemas, incluyendo la pérdida de datos, destrucción de archivos y posibles errores intermitentes en el programa.Todo archivo que se abre debe ser cerrado antes de terminar el programa.

**¿Por qué no se puede abrir un archivo?**

Es posible que el fichero no exista debido a:

•No existe el archivo

•Disco lleno

•Nombre incorrecto

•Directorio no válido

3)1)

r: abre un archivo de texto para lectura

w: Crea un archivo de texto para escritura

a: abre un archivo de texto para añadir

r+: Abre un archivo en modo de lectura y escritura

w+: Crea un archivo en modo de lectura y escritura

a+: añade o crea un archivo de texto para lectura o escritura

rb: Abre un archivo binario en modo de solo lectura.

wb: Crea un archivo binario en modo de escritura

ab: Abre un archivo binario para añadir

r+b: Abre un archivo binario para lectura y escritura

w+b: crea un archivo binario en modo de lectura y escritura

a+b: Añade un archivo binario para escritura y lectura

3) 2) En cuanto a los valores permitidos para los bytes, se puede añadir otro carácter a la cadena de modo:   
"t" se utiliza para indicar que el archivo se abrirá en modo de texto. Este es generalmente el modo por defecto en muchos lenguajes de programación, por lo que a menudo se omite al abrir un archivo. Cuando se abre un archivo en modo de texto, se espera que el contenido sea tratado como texto legible por humanos, y se pueden realizar operaciones de lectura y escritura en el archivo como si fuera un archivo de texto.  
"b" se utiliza para indicar que el archivo se abrirá en modo binario. Al abrir un archivo en modo binario, se espera que el contenido sea tratado como una secuencia de bytes sin procesar en lugar de texto. Este modo es útil cuando se trabaja con archivos que contienen datos binarios, como imágenes, archivos de audio o archivos ejecutables. En modo binario, se pueden realizar operaciones de lectura y escritura de datos sin interpretarlos o realizar conversiones automáticas como ocurre en modo de texto.

3)3)

|  |  |
| --- | --- |
| Función | Significado |
| fopen() | Se utiliza para abrir un archivo |
| fclose() | para cerrar un archivo |
| fgets() | Lee la cadena de un achivo |
| fputs() | escribe la cadena de un achivo |
| fseek() | Busca un Byte especifico de un archivo |
| fprintf() | Escribe la salida con formato del archivo |
| fscanf() | Lee la entrada con formato desde el archivo |
| feof() | Devuelve cierto si se llega al final del archivo |
| ferror() | Devuelve cierto si se produce un error |
| rewind() | Coloca el localizador de posición del archivo el principio del mismo |
| remove() | Borra un archivo |
| fflush() | Vacía un archivo |

3)4)

"FILE" es un tipo de dato estructurado utilizado para representar un archivo. Es una estructura que contiene información sobre el archivo abierto, como su posición, estado y otros detalles necesarios para realizar operaciones de lectura y escritura.

"\*nombrearchivo" es un puntero a una variable de tipo FILE. El puntero se utiliza para hacer referencia a la estructura FILE y acceder a sus miembros y funciones asociadas.

3)5)

Este trozo de programa intenta abrir el archivo "nomfich.dat" en modo de lectura y verifica si la apertura fue exitosa o no. Si no se pudo abrir el archivo, se muestra un mensaje de error. No se menciona específicamente el tipo de archivo al que se está haciendo referencia, pero el nombre del archivo "nomfich.dat" sugiere que podría ser un archivo de datos.

3)6)

1. **FILE \*datosdatos = fopen("nombres.dat", "r");**: El archivo "nombres.dat" se abre en modo de solo lectura ("r"). El puntero FILE "datosdatos" se asigna al archivo abierto.
2. **datos = fopen("nombres.dat", "w");**: El archivo "nombres.dat" se abre en modo de escritura ("w"). Si el archivo ya existe, su contenido se eliminará. Si no existe, se creará uno nuevo. El puntero FILE "datos" se asigna al archivo abierto.
3. **datos = fopen("nombres.dat", "a");**: El archivo "nombres.dat" se abre en modo de anexado ("a"). Si el archivo ya existe, los datos se agregarán al final del archivo. Si no existe, se creará uno nuevo. El puntero FILE "datos" se asigna al archivo abierto.
4. **datos = fopen("nombres.dat", "ra");**: No existe un modo de apertura "ra". La apertura del archivo en este modo resultará en NULL o un comportamiento no definido, ya que no es un modo de apertura válido. Los modos de apertura válidos son "r", "w" y "a".

3)7)

El programa intenta abrir el archivo "c:\banco.dat" en modo de lectura binaria, verifica si la apertura fue exitosa y luego intenta cerrar el archivo, mostrando mensajes de error o éxito según corresponda.