Práctica No. 8 Archivos Binarios

Competencia: *Elaborar programas que utilicen soluciones de software que faciliten el acceso y manejo de información almacenada en arreglos unidimensionales y bidimensionales utilizando la técnica de arreglos paralelos y parámetros por referencia para optimizar el uso de los recursos cuidando además la integridad de la información*.

Un programa durante su ejecución guarda los datos en la memoria RAM, estos datos pueden ser variables, estructuras y arreglos. Los datos también pueden ser enviados a través de flujos a otros lugares como pantalla, impresora y medios de almacenamiento secundario, en forma de archivos.

Un flujo binario puede manejar cualquier tipo de datos, incluyendo datos de texto, pero sin estar limitado a ellos. Los bytes de datos en un flujo binario no son traducidos o interpretados es ninguna forma, sino se leen y escriben tal como son. Los flujos binarios se usan principalmente con los archivos de disco.

El lenguaje C cuenta con varias instrucciones para realizar operaciones con archivos binarios, la secuencia a seguir es más o menos la misma en todas las operaciones:

1. Vincular un archivo de disco con un flujo mediante fopen
2. Realizar cualquiera de las operaciones de escritura, lectura o lectura/escritura.
3. Cerrar el flujo.

fopen

La función fopen, crea un flujo de datos que identifica a un archivo en memoria secundaria y el modo en que opera el flujo. La sintaxis es:

```
FILE * fopen("ruta:/archivo", "modo");
```

FILE* es un apuntador a un archivo. Cuando el archivo no se puede crear por falta de espacio o falta de permisos o por no existir la ruta, el valor del apuntador será NULL.

ruta Es la ubicación del archivo en una unidad de almacenamiento, por ejemplo "c:/tareas/programación/miArchivo". No es necesario especificar una ruta, puede escribirse solo el nombre de archivo y este será creado o buscado en la misma carpeta del proyecto.

modo Es la manera como se abrirá el flujo, puede ser solo para lectura, solo escritura o lectura/escritura. Los modos son:

- wb abre el archive para escritura, si no existe lo crea, si el archivo ya existe lo sobreescribe y se perderá la información anterior.
- ab abre el archivo para agregar al final, si el archivo no existe lo crea.
- rb abre el archive solo para lectura, el archivo debe existir.

rb+ abre el archive para lectura y escritura.

fwrite

La función fwrite escribe un bloque de bytes en un flujo, regresa la cantidad de bytes que han sido grabados. La sintaxis de fwrite es:

```
int fwrite(&estructura, tamaño, cantidad, flujo);
```

&estructura es la dirección de la estructura que contiene los datos.

tamaño Tamaño de la estructura que se va a grabar, este se puede calcular con sizeof(estructura).

Cantidad Cantidad de estructuras que se van a grabar, por lo general, 1**.**

Flujo Variable tipo FILE* que apunta hacia el archivo que se va a grabar.

fread

La función fread lee un bloque de bytes en un flujo y lo guarda en una estructura, regresa la cantidad de bytes que han sido leidos. La sintaxis de fwrite es

```
int fread(&estructura, tamaño, cantidad, flujo);
```

&estructura es la dirección de la estructura que contiene los datos.

tamaño Tamaño de la estructura en donde se van a colocar los datos leidos, se puede calcular con sizeof(estructura).

Cantidad Cantidad de estructuras que se van a leer, por lo general, 1.

Flujo Variable tipo FILE* que apunta hacia el archivo que se va a leer.

ftell

Esta función regresa la posición actual del cursor de lectura/escritura en el flujo. La función regresa -1 si ocurrió un error. El archivo debe existir y debe haberse creado un flujo de lectura o escritura para poder realizar esta operación.

```
long int ftell(flujo)
```

fseek

Esta función sirve para mover el cursor del archivo n bytes a partir de un punto de origen. La función regresa 0 si la función se realizó con éxito y un valor diferente de cero si hubo algún error.

```
int fseek(flujo, desplazamiento, origen);
```

Flujo Variable tipo FILE* que apunta hacia el archivo.

Desplazamiento Cantidad de bytes que se va a desplazar el cursor a partir del origen. Este valor puede ser positivo o negativo.

Origen El origen se puede escribir con valores enteros o con una constante.

0	SEEK_SET	El punto de referencia es el inicio del flujo, el byte 0.
1	SEEK_CUR	El punto de referencia es la posición actual.
2	SEEK_END	El punto de referencia es el final del flujo.

Desarrollo de la práctica

Un diario de circulación local requiere un sistema para controlar el registro de los anuncios económicos de un periódico local. Por cada anuncio se registran los siguientes datos:

- Número de anuncio
- Clasificación (venta varios, autos, mascotas, empleos, renta, traspasos, etc.)
- Responsable de la publicación.
- Teléfono del responsable.
- Contenido del anuncio
- Fecha de publicación

Se pide que el sistema cuenta con los siguientes módulos:

1) Registrar anuncio.
2) Cancelar anuncio (por número de anuncio).
3) Editar anuncio (por número de anuncio).
4) Consulta por clasificación.
5) Consulta por responsable.

6) Ordenar por responsable.
7) Mostrar todos los anuncios que se publicarán en la fecha actual del sistema.
8) Generar respaldo de los datos.
9) Salida

Requisitos:

- El número de anuncio es único, debe verificarse antes de capturar el resto de los datos.
- Debe presentarse un menú para que el usuario elija la categoría del anuncio.
- La cancelación de un anuncio debe hacerse por número de anuncio.
- La edición de un anuncio debe hacerse por número de anuncio.
- Se puede editar cualquier dato excepto el número de anuncio.
- La consulta por categoría debe mostrar todos los anuncios registrados en la categoría que el usuario seleccione.
- La consulta por responsable debe mostrar todos los anuncios registrados con el nombre de responsable que el usuario proporcione.
- Los anuncios deben ser ordenados en forma ascendente según el nombre de responsable.
- La opción mostrar debe presentar en pantalla todos los anuncios que estén registrados para publicarse en la fecha actual del sistema.
- La fecha del sistema debe obtenerse de manera automática.
- La opción generar respaldo de los datos, debe generar un archivo con los datos registrados.
- El nombre del archivo será proporcionado por el usuario.
- Debe presentar mensajes adecuados para cada situación: “No existe”, “Ese número de anuncio ya existe”, “No existe”, etc.
- Debe reutilizar funciones para reducir el tamaño del código.