

# Manejo y Control de archivos

miércoles, 8 de diciembre de 2021 07:15 a. m.

La administración y gestión de archivos nace a partir de la necesidad de resguardar información procesada por algún sistema informático.

La memoria RAM no puede resguardar información cuando se le retira la energía eléctrica( apagado del equipo). Por lo anterior se requiere un mecanismo para resguardar de forma persistente la información y para eso se tienen a las memorias ROM. Sobre las memorias ROM la información ha de guardarse siempre en binario.

El administrador de archivos se encarga de que los sistemas que redirijan la salida estándar hacia algún archivo, lleven los controles de seguridad necesarios.

Se utilizan ACLs(Access Control List), las cuales son reglas que rigen los permisos que tienen los grupos de usuarios, administradores o usuarios sobre un determinado archivo. Los permisos son: lectura, escritura y ejecución.

Los archivos son registros de información en determinado orden y formato. Para manejo conveniente en los lenguajes de programación, todos los formatos se clasifican en dos tipos: texto plano y los binarios.

Las posibles ubicaciones de archivos en cualquier S.O. son: las UBS, los HDD, SDD, penDrivers, Diskets, CD-ROM.

Sintaxis en C:

```
FILE* archivo=NULL;
```

```
archivo =fopen(path, mod);
```

La función fopen() es la que nos permite realizar la apertura de archivos. La función requiere de dos argumentos. El primero es la ruta donde se encuentra el archivo, la cual puede ser relativa o absoluta. El segundo argumento es el modo en el que se realiza la apertura del archivo.

path absoluto: S.O. : windows: C:\\Users\\gusta\\Desktop\\Prueba\_workspace  
\\miLibreria\\tex.txt  
LINUX/UNIX: /home/users/gusta/desktop/text.txt

path relativo: ".\\archivo.txt", "..\\..\\Documentos\\archivo.txt"

Modos de apertura

Modo para texto plano	Modo binario	Descripción
r	rb	Modo Lectura
w	wb	Abre el archivo para escritura, tiene que existir.

a	ab	Abre el archivo para agregar información , sino existe lo crea.
r+	rb+	Abrir un archivo en modo L/E, es archivo posiciona al cursor en el inicio del archivo. Tiene que existir el archivo
w+	wb+	Abre el archivo en modo L/E (sobrescribe el archivo, si no existe lo crea.)
a+	ab+	Abrir el archivo para L/E, pero el cursor se sitúa al final del archivo.

Funciones para el manejo y control de archivos:

Función	¿Qué hace?
rewind(FILE*)	Regresa el cursor a la posición inicial del archivo.
fseek(FILE*, posBytes, CONST)	Mueve el cursor, requiere 3 parámetros: el apuntador al archivo, la cantidad de bytes que se va a mover el cursor, una de las siguientes constantes: SEEK_SET: Cuenta las posiciones desde el principio del archivo. SEEK_CURRENT: Cuenta las posiciones desde la posición actual del cursor. SEEK_END: Cuenta las posiciones de forma regresiva desde la posición final del archivo.
feof(FILE*)	Valida si hemos llegado al final del archivo.
fclose(FILE*)	Cerrar el archivo.

Funciones de escritura

Función	¿Qué hace?
fputc(char, FILE*)	Escribe un carácter en el archivo.
fputs(char*, FILE*)	Escribe una cadena en el archivo.
fprintf(FILE*, "cadenaControl",....);	Funciona de la misma manera que printf(), pero su primer parámetro es el apuntador al archivo. Muy útil en archivos binarios.

Funciones de lectura

Función	¿Qué hace?
fgetc(FILE*)	Lee un carácter del archivo.
fgets(FILE*)	Lee una cadena del archivo.
fscanf(FILE*, "cadenaControl",....);	Funciona de la misma manera que scanf(), pero con un parámetro al principio que indica que se leerá de un archivo.
fread(var, FILE*, tamBytes, int repeticion);	Se utiliza para leer datos binarios de un archivo, recibe 4 parámetros: la variable donde se van a guardar los datos, el tamaño en bytes a leer, cuantas veces se leerá es bloque de bytes y finalmente, el apuntador al archivo.