

Manejo de Archivos

Puntero a la estructura FILE

Para el manejo de archivos es indispensable definir un puntero a la estructura FILE.

```
FILE *pArchivo;
```

Donde 'pArchivo' es el puntero a la estructura FILE.

Apertura de un archivo

Cada vez que se necesite trabajar con un archivo , el primer paso es abrirlo. Si no se realiza esto no se puede leer ni escribir en el mismo. La función que permite la apertura del archivo es 'fopen()'.

```
FILE * fopen (const char *Path_al_archivo , const char *Modo)
```

Modos de Apertura

Modo	Descripción
"r"	abrir un archivo para lectura, el fichero debe existir.
"w"	abrir un archivo para escritura, se crea si no existe o se sobrescribe si existe.
"a"	abrir un archivo para escritura al final del contenido, si no existe se crea.
"r+"	abrir un archivo para lectura y escritura, el fichero debe existir.
"w+"	crear un archivo para lectura y escritura, se crea si no existe o se sobrescribe si existe.
"rb+"/ "rb"	Abre un archivo en modo binario para actualización (lectura y escritura).
"rb"	Abre un archivo en modo binario para lectura.
"wb"	Abre un archivo en modo binario para escritura.

Cierre de un archivo

Todo archivo que se abre debe ser cerrado antes de terminar el programa. Terminar el programa sin cerrar el o los archivos puede causar pérdida de datos. La función 'fclose()' es la que se encarga de cerrar un archivo.

```
int fclose ( FILE *pArchivo );
```

Si el archivo es cerrado exitosamente se retorna un 0 , en caso contrario se devuelve -1;

```
#include <stdio.h>
void main (void)
{
    FILE *pArchivo;
    pArchivo = fopen("banco.txt","rb");
    if(pArchivo==NULL)
    {
        printf("\nEl archivo no puede ser abierto");
        exit (1);
    }
    fclose(pArchivo);
}
```

Escritura de un archivo

Una vez que el archivo se encuentra abierto se puede empezar a trabajar para leer o escribir. La función utilizada para realizar la escritura es `fwrite`. Esta función sirve para escribir archivos de texto o binarios.

```
int fwrite ( void * origen , size_t tamaño , size_t cantidad , FILE *arch);
```

Devuelve el número de datos escritos (cantidad). Si el valor retornado es menor al que se indicó por medio de la variable cantidad , significa que hubo un error en la escritura.

Escritura de un archivo

```
int fwrite ( void * origen , size_t tamaño , size_t cantidad , FILE *arch);
```

- **origen**: Es un puntero al lugar desde donde se obtienen los datos para escribir en el archivo

Escritura de un archivo

```
int fwrite ( void * origen , size_t tamaño , size_t cantidad , FILE *arch);
```

- origen: Es un puntero al lugar desde donde se obtienen los datos para escribir en el archivo
- tamaño: Es el tamaño en bytes del dato que se va a escribir

Escritura de un archivo

```
int fwrite ( void * origen , size_t tamaño , size_t cantidad , FILE *arch);
```

- origen: Es un puntero al lugar desde donde se obtienen los datos para escribir en el archivo
- tamaño: Es el tamaño en bytes del dato que se va a escribir
- cantidad: Es la cantidad de datos de longitud tamaño que se van a escribir

Escritura de un archivo

```
int fwrite ( void * origen , size_t tamaño , size_t cantidad , FILE *arch);
```

- origen: Es un puntero al lugar desde donde se obtienen los datos para escribir en el archivo
- tamaño: Es el tamaño en bytes del dato que se va a escribir
- cantidad: Es la cantidad de datos de longitud tamaño que se van a escribir
- arch: Es el puntero a FILE asociado al archivo

```
int cantidadEscrita , longitudTexto;  
FILE *pArchivo;  
char texto[ ] = "hola mundo!!!";  
pArchivo = fopen("banco.txt","w");  
if (pArchivo == NULL) {  
    printf("\nError de Apertura");  
    exit(1);  
}  
longitudTexto = strlen (texto);  
cantidadEscrita = fwrite(texto ,sizeof (char), longitudTexto , pArchivo );  
if (cantidadEscrita<longitudTexto)  
    printf("\nError al escribir el archivo");  
  
fclose(pArchivo);
```

ARCHIVOS DE TEXTO

```
struct Persona { char nombre[10]; int edad; };  
struct Persona auxiliar;  
int cantidadEscrita;  
FILE *pArchivo;  
strcpy(auxiliar.nombre, "JUAN");  
auxiliar.edad = 20;  
pArchivo = fopen("banco.bin","wb");  
if (pArchivo == NULL)  
    exit(1);  
cantidadEscrita=fwrite(&auxiliar, sizeof(struct Persona), 1, pArchivo);  
if (cantidadEscrita < 1)  
    printf("\nError al escribir el archivo");  
  
fclose(pArchivo);
```

ARCHIVOS BINARIOS

Lectura de un archivo

Para realizar la lectura de un archivo se utiliza la función `fread()`. Esta función sirve para leer archivos de texto o binarios.

```
int fread( void * destino , size_t tamaño , size_t cantidad , FILE *arch);
```

Devuelve el número de datos leídos (cantidad).

Lectura de un archivo

```
int fread ( void * destino , size_t tamaño , size_t cantidad , FILE *arch);
```

- **destino**: Es un puntero al lugar donde se va a dejar el dato leído con fread()
- **tamaño**: Es el tamaño en bytes del dato que se va a leer
- **cantidad**: Es la cantidad de datos de longitud tamaño que se van a leer
- **arch**: Es el puntero a FILE asociado al archivo

Lectura de un archivo

```
int fread ( void * destino , size_t tamaño , size_t cantidad , FILE *arch);
```

- destino: Es un puntero al lugar donde se va a dejar el dato leído con fread()
- tamaño: Es el tamaño en bytes del dato que se va a leer
- cantidad: Es la cantidad de datos de longitud tamaño que se van a leer
- arch: Es el puntero a FILE asociado al archivo

Lectura de un archivo

```
int fread ( void * destino , size_t tamaño , size_t cantidad , FILE *arch);
```

- destino: Es un puntero al lugar donde se va a dejar el dato leído con fread()
- tamaño: Es el tamaño en bytes del dato que se va a leer
- cantidad: Es la cantidad de datos de longitud tamaño que se van a leer
- arch: Es el puntero a FILE asociado al archivo

Lectura de un archivo

```
int fread ( void * destino , size_t tamaño , size_t cantidad , FILE *arch);
```

- destino: Es un puntero al lugar donde se va a dejar el dato leído con fread()
- tamaño: Es el tamaño en bytes del dato que se va a leer
- cantidad: Es la cantidad de datos de longitud tamaño que se van a leer
- arch: Es el puntero a FILE asociado al archivo

ARCHIVOS DE TEXTO

```
int cantidadLeida , longitudTexto;
```

```
FILE *pArchivo;
```

```
char texto[50] ;
```

```
pArchivo = fopen("banco.txt","r");
```

```
...
```

```
longitudTexto = 50;
```

```
while(!feof(pArchivo))
```

```
{
```

```
    cantidadLeida = fread(texto,sizeof (char), longitudTexto , pArchivo );
```

```
    printf("El texto leido es: %s", texto) ;
```

```
}
```

```
fclose(pArchivo);
```

```
struct Persona { char nombre[10]; int edad; };
```

```
struct Persona auxiliar;
```

```
int cantidadLeida;
```

```
FILE *pArchivo;
```

```
pArchivo = fopen("banco.bin","rb");
```

```
...
```

```
while(!feof(pArchivo))
```

```
{
```

```
    cantidadLeida = fread(&auxiliar, sizeof (struct Persona), 1 , pArchivo );
```

```
    if( cantidadLeida == 1)
```

```
        printf("Nombre: %s - Edad: %d", auxiliar.nombre, auxiliar.edad) ;
```

```
}
```

```
fclose(pArchivo);
```

ARCHIVOS BINARIOS