

archivos

crear un archivo:

la funcion **fopen** recibe 2 parametros:

1. el nombre del archivo en disco que quiero abrir. en linux no es tan necesario especificar la extension, y va entre comillas dobles.
2. este parametro de uno o dos caracteres, especifica el tipo de operaciones que quiero hacer con el archivo.
algunas de las opciones para el segundo parametro son:
 - r) solo lectura: cursor al principio
 - r+) lectura y escritura: cursor al principio
 - w) crea un nuevo archivo o resetea a vacío el existente: cursor al principio
 - w+) igual a w pero permite leer
 - a) agrega escritura al final del archivo o crea uno nuevo si no existe, cursor al final.
 - a+) agrega escritura al final y permite leer. en ubuntu el cursor va al principio, en mac, android y BSD al final
 - rb) leer binario
 - wb) escribir binario
 fopen devuelve un descriptor al archivo,

```
FILE *myFile = fopen("datos.bin", "rb");
if (myFile == NULL)
{
    printf("\n=====cannot open file\n=====");
    return 1;
}
printf("\n=====success fopen\n=====");

/* aqui usaremos fread y fwrite, manejaremos los datos obtenido y demas.
luego es necesario cerrar el archivo que hemos abierto */

printf("\n=====closing file\n=====");

int closeStatus = fclose(myFile);

closeStatus != 0 ? printf("error closing file (%i)\n=====",
closeStatus) : printf("success closing file (%i)\n=====",
closeStatus);
```

para cerrar el archivo usamos la funcion **fclose** que recibe simplemente el descriptor del archivo abierto e intenta cerrarlo, si devuelve cero, se cerro correctamente.

es necesario cerrarlo para que otros programas puedan acceder al archivo

leer el contenido del archivo

para comezar usaremos **fgetc**, que recibe el descriptor del archivo abierto, y devuelve un entero que representa el ASCII del caracter que esta leyendo el *cursor*. el cual por ejemplo podriamos castear a char para imprimirlo por pantalla:

```
int charLeido = fgetc(myFile);  
printf("he leído el caracter %c\n", (char)charLeido);
```

para evitar errores como por ejemplo intentar leer estando al final del archivo, podemos aprovechar el caracter EOF *end of file* y la funcion **feof**, la cual recibe el descriptor del archivo abierto y devuelve false o cero si el caracter actual es *distinto de EOF* y en caso de que sea EOF, devuelve true, o no-cero.