Trabajo de investigación de archivos

Buscar, definir y ejemplificar los siguientes conceptos adjuntando las

imágenes correspondientes

1. Crear un archivo en c : https://es.wikibooks.org/wiki/Programaci%C3%B3n\_en\_C/Manejo\_de\_archivos

En el campo Archivo fuente, escriba el nombre del archivo que se creará, incluida la extensión. Pulse Finalizar. El archivo se abrirá automáticamente en el editor C/C++ remoto. Especifique el código en el editor y guarde el archivo.

#include <stdio.h>

int main() {

// Abre un archivo llamado "archivo.txt" en modo de escritura (sobrescribe si ya existe)

FILE \*archivo;

archivo = fopen("archivo.txt", "w");

// Verifica si se abrió correctamente el archivo

if (archivo != NULL) {

// Escribe contenido en el archivo

fprintf(archivo, "Hola, este es un archivo de ejemplo creado en C.\n");

fprintf(archivo, "Puedes escribir cualquier cosa que desees aquí.\n");

// Cierra el archivo

fclose(archivo);

printf("Se ha creado el archivo correctamente.\n");

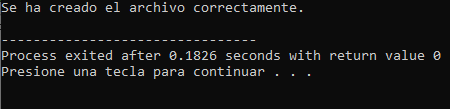
} else {

printf("Error al crear el archivo.\n");

}

return 0;

}



1. Abrir y cerrar un archivo

FOPEN ;

Esta función sirve para abrir y crear ficheros en disco. El prototipo correspondiente de **fopen** es:

1. FILE \* fopen (**const** char \*filename, **const** char \*opentype);

FCLOSE;

Esta función sirve para poder cerrar un fichero que se ha abierto. El prototipo correspondiente de **fclose** es:

int fclose (FILE \*stream);

1. Leer, escribir, sobre escribir, etc. (parámetros)

Un archivo generalmente debe verse como un string (una cadena de caracteres) que esta guardado en el disco duro. Para trabajar con los archivos existen diferentes formas y diferentes funciones. Las funciones que podríamos usar para leer un archivo son:

* char fgetc(FILE \*archivo)
* char \*fgets(char \*buffer, int tamano, FILE \*archivo)
* size\_t fread(void \*puntero, size\_t tamano, size\_t cantidad, FILE \*archivo);
* int fscanf(FILE \*fichero, const char \*formato, argumento, ...);

Así como podemos leer datos desde un fichero, también se pueden crear y escribir ficheros con la información que deseamos almacenar, Para trabajar con los archivos existen diferentes formas y diferentes funciones. Las funciones que podríamos usar para escribir dentro de un archivo son:

* int fputc(int caracter, FILE \*archivo)
* int fputs(const char \*buffer, FILE \*archivo)
* size\_t fwrite(void \*puntero, size\_t tamano, size\_t cantidad, FILE \*archivo);
* int fprintf(FILE \*archivo, const char \*formato, argumento, ...);

Para sobreescribir solo abrimos el archivo nuevamente y con fprintf sobreescribimos sobre el :

fprintf(archivo, "Este es el nuevo contenido.\n");

d) Moverse dentro del fichero y escribir la palabra “PRINCIPIO” ,

“ACTUAL” O “FINAL” según corresponda.

#include <stdio.h>

int main() {

// Abre un archivo llamado "archivo.txt" en modo de lectura y escritura

FILE \*archivo;

archivo = fopen("archivo.txt", "r+");

if (archivo == NULL) {

printf("Error al abrir el archivo.\n");

return 1; // Sale del programa con un código de error

}

// Obtiene la posición actual en el archivo

long int posicion\_actual = ftell(archivo);

if (posicion\_actual == -1L) {

printf("Error al obtener la posición actual en el archivo.\n");

fclose(archivo);

return 1; // Sale del programa con un código de error

}

// Mueve el puntero de archivo al principio del archivo

fseek(archivo, 0, SEEK\_SET);

// Escribe la palabra "PRINCIPIO"

fprintf(archivo, "PRINCIPIO ");

// Mueve el puntero de archivo a la posición original

fseek(archivo, posicion\_actual, SEEK\_SET);

// Escribe la palabra "ACTUAL"

fprintf(archivo, "ACTUAL ");

// Mueve el puntero de archivo al final del archivo

fseek(archivo, 0, SEEK\_END);

// Escribe la palabra "FINAL"

fprintf(archivo, "FINAL");

// Cierra el archivo

fclose(archivo);

printf("Se ha escrito en el archivo correctamente.\n");

return 0;

}

1. Diferencia entre fputc y fputs

### FPUTC :

Esta función escribe un carácter a la vez del archivo que esta siendo señalado con el puntero **\*archivo**.

FILE \*archivo;

archivo = fopen("archivo.txt", "w");

fputc('A', archivo);

fclose(archivo);

**FPUTS :**

La función **fputs** escribe una cadena en un fichero. la ejecución de la misma no añade el carácter de retorno de línea ni el carácter nulo final. El valor de retorno es un *número no negativo* o **EOF** en caso de error. Los parámetros de entrada son la cadena a escribir y un puntero a la estructura **FILE** del fichero donde se realizará la escritura.

FILE \*archivo;

archivo = fopen("archivo.txt", "w");

fputs("Hola mundo", archivo);

fclose(archivo);

La principal diferencia es que **fputc** se usa para escribir caracteres individuales, mientras que **fputs** se usa para escribir cadenas de caracteres.

Nota: el trabajo debe estar perfectamente referenciado