

INFORMÁTICA II

TRABAJO PRÁCTICO N° 4

Archivos

Recomendaciones:

- En este TP puede usar e incluir las librerías de `c string.h`, `file.h`, `stdio.h`, `stdlib.h`
- Recuerde validar si el archivo pudo ser creado/abierto, antes de continuar con el resto del programa.
- Recuerde, al finalizar de manipular un archivo (leerlo, escribirlo, etc), cerrar el archivo.

EJERCICIO 1

El nombre del archivo lo deberá ingresar el usuario. Ud. debe crear una función que se encargue de validar el nombre de archivo ingresado. Puede usar funciones de la librería `string.h`

Debe verificar que el nombre sólo tenga letras (mayúsculas o minúsculas) y/o números, más la extensión del archivo.

Nombre del archivo —> `ejemploCodigoFuente.c` ← extensión del archivo

Ejemplos de nombres válidos: `t3xt0.txt` , `codigo4.cpp`, `tp4.c`, `misNotas.txt`

Ejemplos de nombres inválidos: `t3xt0` (no tiene extensión de archivo), `tp4-2.cpp` (contiene un guión), `h.` (olvidó la extensión), `.cpp` (olvidó el nombre), `h*la.txt` (contiene caracter asterisco).

Luego del ingreso del nombre de archivo y su validación, deberá permitir que ingrese por pantalla texto y luego almacenarlo en un archivo cuyo nombre ya eligió el usuario.

Posteriormente, partiendo del archivo cerrado, se deberá leer el contenido del archivo y mostrarlo en pantalla.

Si se vuelve a abrir el mismo archivo (se ingresa el mismo nombre de archivo), deberá añadirse el nuevo texto al final del archivo sin borrar su contenido anterior (modo append “a” de apertura de archivo).

EJERCICIO 2

Realizar un programa que permita copiar archivos. El programa deberá abrir el archivo “Origen” y crear un nuevo archivo “Destino” el cual resultará una copia exacta del primero en tamaño y contenido.

El nombre del archivo “Origen” y su extensión lo ingresa el usuario. El nombre del archivo “Destino” es de su elección, pero recuerde que DEBE mantener la extensión de archivo.

Pruebe la ejecución con distintos tipos de archivo (.txt, .cpp, .jpg u otros de imágenes)

EJERCICIO 3

Debe tener al menos un archivo de texto con contenido ya creado. A modo de ejemplo en esta consigna usamos Origen y Destino para referirse a los archivos de texto a usar.

Realice un programa que permita el ingreso de los dos nombres de archivo: uno Origen y otro Destino.

Debe verificar que ambos archivos existan o estén disponibles y sean nombres válidos.

Luego debe permitir el ingreso de dos palabras.

Se deberá copiar el contenido del archivo Origen en el Destino, **pero** reemplazando toda palabra que se encuentre igual a la primera ingresada, por la segunda ingresada.

EJERCICIO 4

Copie el código fuente de su ejercicio 2 del TP3 de estructuras. Añada a su programa la funcionalidad para guardar en archivo el arreglo de estructuras y para mostrarlo luego. Use las funciones fread y fwrite para la lectura y escritura del archivo binario.

```
size_t fread(void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

```
size_t fwrite(const void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

Comente su código fuente (usando // o /* */) para aclarar la funcionalidad de su programa.

EJERCICIO 5

Copie el código fuente de su ejercicio 3 del TP3 de estructuras. Añada a su programa la funcionalidad para guardar en archivo el arreglo de estructuras y para mostrarlo luego. Use las funciones fread y fwrite para la lectura y escritura del archivo binario.

```
size_t fread(void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

```
size_t fwrite(const void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

Comente su código fuente (usando // o /* */) para aclarar la funcionalidad de su programa e indicar si guarda ambas funciones en un mismo archivo o en archivos separados.

EJERCICIO 6

Copie el código fuente de su ejercicio 4 del TP3 de estructuras. Añada a su programa la funcionalidad para guardar en archivo el arreglo de estructuras y para mostrarlo luego. Use las funciones fread y fwrite para la lectura y escritura del archivo binario.

```
size_t fread(void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

```
size_t fwrite(const void *ptr, size_t size_of_elements, size_t number_of_elements, FILE *a_file);
```

Comente su código fuente (usando // o /* */) para aclarar la funcionalidad de su programa.

Referencias

Videos de teoría en el campus virtual sobre estructuras, cadenas y archivos.

Código ejemplo subido en el campus virtual y sus tutoriales.

Manejo de cadenas de caracteres en C (en inglés
https://en.wikipedia.org/wiki/C_string_handling

Programación en C, wikilibro,
https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Manejo_de_archivos