

Documentación

Explicación general-----pág. 1

Warnings y errores-----pág. 2

Aclaraciones-----pág. 3

Explicación general

El programa consiste de un menú con 10 opciones:

- 0-Salir.
 - Sale del programa.
- 1-listar_txt
 - Imprime por pantalla el archivo csv con el nombre prestamos.csv (tiene que estar en la misma carpeta que el programa).
- 2-crear_dat
 - Crea un archivo binario con el nombre “creditos.dat”.
 - Si ya existe lo vacía.
- 3-migrar
 - Migra los datos del archivo “prestamos.csv” a “creditos.dat”, si este último tenía datos los sobrescribe.
- 4-listar
 - Imprime por pantalla el contenido del archivo “creditos.dat”. Puedo elegir si me imprime los activos, todos, según el tipo de crédito o los que están dentro de un rango de fechas.
- 5-alta
 - Me permite dar de alta un cliente en el archivo binario.
- 6-buscar
 - Puedo buscar un cliente determinado, ya sea por apellido o por número de orden.
- 7-modificar
 - Puedo modificar algunos datos de un cliente que yo elijo.
- 8-Baja lógica
 - Puedo dar de baja un cliente. Los datos del cliente dado de baja se pasan a un archivo de texto con terminación “.xyz” que se llama “clientes_bajas_<fecha>.xyz”
- 9-listar xyz
 - Imprime por pantalla los registros en el archivo “clientes_bajas_<fecha>.xyz”.

El archivo tiene 4 librerías: estructuras.h (donde están los structs), archivos.h (donde están las funciones para trabajar con archivos), validaciones.h (donde están las funciones para validar y algunas funciones auxiliares) y menu.h (donde están las funciones para imprimir el menú). Cada una con sus respectivos .c.

Apenas se ejecuta el programa se lee el archivo “prestamos.csv” y se crea una copia con los ‘;’ cambiados por ‘,’ y las ‘,’ cambiadas por “.”. Por lo tanto, asegúrese que el archivo csv se encuentre en la misma carpeta que el código.

A su vez también se fija si existe un archivo binario “creditos.dat” de existir lo lee y lo carga a memoria. Revisar que en la misma carpeta del código no haya ningún archivo con ese nombre porque puede llegar a ser reescrito.

Warnings y errores

En cuanto a errores o ‘bugs’ sólo encontré 1, al ejecutar la opción 9 “listar xyz” y si no se realizó ninguna baja lógica en esa sesión del programa, se va a levantar un error el cual dirá que no se encontró el archivo y cerrará el programa. Esto se debe a que el nombre del archivo xyz no está cargado en memoria y no lo encuentra.

Considero que no es muy difícil de corregir, pero no llego por el tiempo.

En cuanto a warnings hay alrededor de 8, la mayoría son de tipos “incompatible” cuando paso parámetro o en algún printf, pero ninguno da problemas y tampoco logré solucionarlos. Hay alguno también de “implicit declaration” con alguna función que invoco antes de declarar el cuerpo en ese mismo archivo, pero tampoco me dio problemas.

Aclaraciones

Esta parte va dirigida a /los **profesores** encargados de corregirme.

El programa utiliza pathing de Windows para incluir algunas librerías, por lo que de ejecutarse en otro sistema operativo podría llegar a dar problemas. No llegué a testearlo ni en Linux ni en macOS.

Pude resolver todos los puntos de una manera que yo considero eficaz, pero que puede no ser eficiente debido a que la mayor parte del trabajo lo realicé cargando los datos de los archivos en un array el cual está presente en memoria durante la ejecución del programa. Si este archivo llegase a tener miles o cientos de miles de registros sospecho que ralentizaría la máquina y lo volvería un programa muy poco eficiente.

Aun así, me di cuenta de este posible problema tarde, y ya no me queda tiempo para cambiarlo, pero para la cantidad de registro que tiene actualmente y para los que puede llegar a tener en un futuro cercano considero que este programa cumple su función de una manera decente en términos de eficiencia.