SS

|  |  |
| --- | --- |
| Cinema Management Application  © 2018 | Nikolaos Perris  [36261@ufp.edu.pt](mailto:36261@ufp.edu.pt)  Engenharia Informática  Universidade Fernando Pessoa  Project work for Introduction to Algorithms and Programming |

**Summary**

As asked by you, this application is used to manage a chain of cinemas.

It can manage the data related to the movies available in the cinemas and also manage the actors that participated in those movies.

**Installation Guide**

To run the program, you simply need to run the CinemaManagement.exe executable included in the zip file. You can use the load database option to load existing sample data or create your own.

**User Manual (not updated for v0.97)**

In the main menu, there are options to (1) manage cinemas, (2) manage movies, (3) manage actors, (4) manage movie genres, (8) load database, (9) save database and (Q) quit the program. The first 4 options open new sub-menus. Option (8) allows the user to load the previously saved data from a file. Option (9) allows the user to save the data to a file. Option (Q) is used to exit from the program.

Sub-menu (1) for the cinemas, has options to (1) list the cinemas, (2) list the movies that are being played in a cinema, (3) list the cinemas that a movie is available in, (4) list the movies of a genre and the cinemas they are available in, (5) associate movies to cinemas, (8), load cinemas data from file, (9) save cinemas data to file. Option (R) allows the user to return to the main menu.

Sub-menu (2) for the movies, has options to (1) create a new movie entry, (2) edit a movie entry, (3) delete a movie entry, (4) associate actors to a movie, (5) associate movies to cinemas (same as 1.5), (6) list the movies by title, (7) list the movies with all their information, (8) load the movies data from file, (9) save the movies data to a file. Option (R) allows the user to return to the main menu.

Sub-menu (3) for the actors has options to (1) create a new actor entry, (2) edit an actor entry, (3) delete an actor entry, (4) associate three actors to a movie, (5) list the actors by name, (6) list the actors with all their information, (7) list the movies an actor has participated in, (8) load the actors’ data from an existing file, (9) save the actors data to a file. Option (R) allows the user to return to the main menu.

Sub-menu (4) for the movie genres has options to (1) create a new genre entry, (2) edit a genre entry, (3) delete a genre entry, (4) list all the available genres, (5) list the movies of a genre and the cinemas they are available in (same as 1.4), (8) load genres data from a file, (9) save the genres data to a file. Option (R) allows the user to return to the main menu.

Error messages are displayed whenever the user makes an invalid option.

**Technical comments**

To make this program, I used the as a starting point the “Musicas” program we saw during classes. I tried to give a decent look to the program and make it easy to navigate through the menus and options.

The available cinemas are 5 and are included in a pre-defined array, while movies and actors are separate structures that contain data the user can create, edit and delete. Genres are stored in an array. Structure for cinemas includes 4 movies that are saved there after doing the associations. Structure for movies includes index, title, director, genre, duration, minimum age and three actors that are saved there after doing the associations. Structure for actors includes an index, the name and the age of the actor. Indexes are automatically generated and not entered by the user when creating data entries.

Menus use the switch function and the user can type the number of the option he wants. I used getch() function instead of scanf() to not have to wait for ENTER key after pressing the option key. If a wrong key is pressed while in the menus, a message is displayed and an alert sound is played. I read online that getch() is a Windows only function, so that could cause problems on other platforms.

The program uses text files to save the data at this moment, as asked by you, but I want to try using binary files too. The functions to create new movies and actors are based on the “Musicas” functions and working as intended. The functions to edit and delete movies/actors are working by displaying a list with the existing movies or actors, letting the user to choose the number of the movie/actor he wants to edit or delete. The delete functions work by moving one position to the left, all elements to the right of the element the user selected. That way, the element the user selected, gets overwritten, so it’s effectively deleted.

The edit functions I had trouble to make them work with the use of gets() like in the create function, so I used scanf() function and the subset %[^\n] to be able to read the space characters and save them. I found this solution by searching online. I get some compiler warnings but it seems to work. For each entry, I first display the old data, then I prompt for the new data.

Instead of displaying lists I want to try changing the functions so they work by letting the user to type the full name of movie or actor, checking if there is a matching string in the database and if yes, letting the user to edit/delete them.

All the save functions first check if there is data to save. If there isn’t, a message is displayed and the program does not save anything, to avoid overwriting the existing files with empty files. The data files movies.txt, actors.txt, cinemas.txt and genres.txt I am using to test the functions of this program use data from the current Top Box Office (US), as found in [www.imdb.com](http://www.imdb.com) . They contain 10 movies and 3 actors from each movie, total 30 actors (actually 29, one was duplicate and I deleted). For cinemas, I use the names of 5 NOS cinemas in Porto.

A copy of these files is in the test files subfolder to be able to quickly restore them in case of making changes to the files of the root folder.

I have tried to include lots of comments in the source code to help you understand what each part of code does. All testing was done to my desktop and laptop computers, using Windows 10 and Windows 8.1.

The limitations of the program are:

* The cinemas are pre-defined. I could change that but if I understood correctly it was not required to create and edit cinemas data. **<= FIXED**
* A cinema can have a max of 4 movies associated. **<= FIXED**
* A movie can have a max of 3 actors associated. **<= FIXED**
* At this moment, when creating a movie entry, the genre is entered manually and not selected from the existing ones, so it could cause problems like having “Sci-fi” in genres data and “Sci-Fi” in movies data, or not having a genre at all.

**Extra comments**

I am not yet very comfortable working with structures, arrays and especially pointers, so my code is most likely too long and not very efficient. I am sure for several functions I could use shorter code but due to limited time, my goal was to find a working solution, but not necessarily the best solution.

I only found out about this project work, really late, around start of December, as I told you in class. I had no time to work on it during December, as I had several other exams and then went to Greece, so I only started working on it 2 weeks ago. Also, as I had to work alone, there is a possibility that I have misunderstood something from your document that described the work we should do, which was written only in Portuguese. I hope that did not happen.

Also, I uploaded on 13/01/2018 a version that the association functions were not implemented. Later I received an email that we could deliver the work until 14/01/2018 so I managed to implement those functions before the deadline. Copies of old main.c and current main.c are in the root directory.

**Todo list:**

* Remove the limits of 4 movies per cinema and 3 actors per movie and change the associate functions. **<= DONE**
* Improve the duplicate checks. **<= DONE**
* Confirmations for edits/deletions. **<= DONE**
* Choose a genre from the existing ones when creating a movie.
* Any char can be used for admin password, including control chars. I want to limit the available chars. **<= DONE**
* Make a better encryption of the admin password that is saved in the binary file.
* Use binary files for all save/load functions.

**Changelog:**

15/01/2018, v0.96:

* Fixed a bug in the list\_cinemas\_of\_movie and list\_movies\_by\_genre functions, by using strcmp function, to properly compare.
* Added a check if the new entries are duplicated. It only checks after entering all data of a movie or actor, not just the name, so it needs improvement.
* Minor cosmetic changes (added my name under the program name on the menus, some changes in the messages, etc).
* Added menu\_title, invalid\_selection functions to shorten the code a bit.
* Added input validation for edit/delete genre functions.
* Added confirmation question for save operations.
* Added confirmation question for load operations, when there is already data in the memory. If there is no data, it doesn’t ask. Added some more comments in the source code.

17/01/2018, v0.97

* Added user selection menu as first screen. User menu has view-only options. Administrator menu has options to manage everything. There is also a load database option in user selection menu.
* Added password protection for administrator menu. Default admin pass is ‘admin’ and there is a secret developer pass that can also be used to enter the admin menu (not so secret for you, it’s in the source code!).
* Added option to change admin password. It accepts any char, including control chars. Max length 20 chars. Restriction: you can’t use backspace if you make a mistake, as it accepts it as a char too.
* Added a small loop to display asterisks when typing passwords.

Loop taken from <http://codingstreet.com/c-code-to-hide-input-password/>

* After changing default admin password, new password is saved in a binary file topsecret.dat and is loaded when you start again the program, making the default pass ‘admin’ to not work anymore. (the developer pass still works ofc). Added a little trick to ‘encrypt’ the password in the file, so it’s not directly readable.

20/01/2018, v0.98

* Major changes in the cinemas structure.
* Cinema names are no longer predefined. There are now options to create and delete cinemas in the cinemas menu.
* There is no limit anymore when assigning movies to cinemas: user can select the number of the movies and it can be different for each cinema. That number is saved in the structure for use in other functions.
* Adjusted several functions to work with the new structure of the cinemas. (save/load, associations, etc)
* Added question for user confirmation to all edit/delete functions.
* Lots of small cosmetic changes, improvements to the displayed messages,etc.

21/01/2018, v0.99

* Major changes in the movies structure.
* There is no limit anymore when assigning actors to movies: user can select the number of the actors and it can be different for each movie. That number is saved in the structure for use in other functions.
* Adjusted several functions to work with the new structure of the movies. (save/load, associations, etc)
* Minor tweak in edit function to get rid of compiler warnings (using gets instead of scanf with subset). I now get 0 warnings.
* Improved duplicate checks when creating cinemas, movies, actors, genres.
* Added duplicate checks when editing cinemas, movies, actors, genres. If duplicates are found in other entries, no changes are made. If the new name is same as the old or different(but not duplicate), it proceeds normally to the rest of the edit function.

22/01/2018, v0.99a

* Changed all gets() to fgets() to limit the input to the available, cause bigger than NAME\_SIZE inputs could lead to crashes in several functions. Used fflush(stdin) to discard the extra input before the next fgets.
* Added duplicate checks to the association functions.
* When associating movies to cinemas and actors to movies, the selections are added to the ones that had been already associated before, if any. Previously, it was just associating from scratch every time, overwriting any existing associations.
* Added options to delete and edit associations, with duplicate checks, etc.
* Added new Manage Associations menu, which is under the admin menu and contains all the associations related options.

24/01/2018, v0.99b

* Small fix in admin pass functions. Backspace can now be used to delete user input, as intended.