Project 02 – mp3 shuffler

Marks: 10% of the final grade (zero marks for submissions that have compilation errors)

Deadline: strictly on or before May 08th 2015 @ 5.00PM. Late submissions **will not** be accepted.

Plagiarism policy: Copied submissions (including those from the Internet) will receive zero marks. Your program must be ENTIRELY your own works. DO NOT COPY FROM OTHERS and DO NOT ALLOW ANYONE ELSE TO SEE YOUR CODE.

Task:
Idea of this project is to develop an mp3 shuffler: that is an mp3 player that would play a given set of songs in a random or shuffled manner. We call this *player* hereafter.

Description:
For playing the mp3 songs you should use a popular player called *mplayer* which can be invoked from the command line. While it is possible to use other players as well you **should** use *mplayer* in this project since it would make marking the project easy. Further *mplayer* is installed in all the CE top floor lab machines.

Reading songs: You can download some sample code together with few mp3 songs from the FEeLS. Your player should first read the given directory and from that figure out what are the mp3 songs that are in that directory. For simplicity you can assume that you do not have anything other than mp3 songs in that particular said directory. The directory name should be read from the command line;
$ player song_directory. If you do not give a directory name via the command line then the payer should look for mp3 songs in the current working directory (or ./).

You should understand the given sample code in *readfiles.c* which would help you to achieve this task.

Invoking the *mplayer*: The *mplayer* can be invoked via your program using the *system* function provided by the C programming language. You can see the sample code in *runinshell.c*. Here note that you need to generate the command into a single string and pass to the system function.

Warning: When you pass **file names** into the shell and hence the *system* command, they **cannot** contain spaces. Of cause mp3 file names will contain spaces; so you need to make these names *shell proof* by putting '\' (escape) before each and every space.

Shuffler: The basic idea of shuffling is to play the given set of files in a random order. However it is bit complicated than that. Suppose you are asked to guess a random number between 1 and 10. If you repeat this guessing for say 10 times then there is a good chance you may get the same number twice. (You can examine and run the code in *randnumber.c* to understand this issue). That means you will play the same song twice which is not what you expect from a shuffler. So your shuffler should make sure that it will not play the same song twice.

Summary of the specifications:

- Should play all the mp3 songs in the specified directory.
- The directory should be specified via the command line and if no directory is specified you should search in current working directory.
- You may assume all files in the said directory are mp3.
- You cannot make any assumptions about the number of songs in the directory.
- Song names will have spaces which cannot be directly handled by the shell.
- You should play a random order of the given set of songs.
- You should make sure the same song is not played twice.

Hints:

Some commands/web sites that might help

- The man command can be used to get the manual page of any command; for example type *man system*
- https://www.cs.bu.edu/teaching/cpp/string/cstring/ has a good introduction about strings in C.

Submission:

You SHOULD submit a single C file with the following filename: **player.c** before the deadline. Files with any other filename will NOT be considered for marking.

Note that, marks will also be awarded for legible and readable code, good coding practice and proper comments.

GOOD LUCK and ENJOY CODING.