

Quipper

Movie Quote generator written in C.

Link to GitHub repository: <https://github.com/cdc67/Quipper>

By:

- Chris Callans
- Stephen Calabro
- Ben Greenfield

Overview

Quipper is a movie quote generator written in C. It provides functionality to gain quotes from random movie sources, or to lookup quotes for a specific movie by title. Quipper uses IMDB to gain all of the information for the movies.

Features

- List movie quotes by Title
- Randomly select movie quotes from a list
- Run as a background process

Usage

To use this program. Download all of the files that are in this repository to a location on your computer. Navigate to that location using a terminal program. Once inside simply type make, or make all. To run the program. This will then bring you to the main ui of the program.

Note: You may also just run the following command to build the program (make build), and then run it as any other executable file (./quipper)

Documentation

This is all of the documentation for the files listed in the program.

quipper.c

struct movieList

Wrapper for a resizable array: char** movies - List of movie titles int size - Current number of titles in movies int cap - Current capacity of movies

`int Search_in_File(char* fname, char* title, int bg)`

This function searches through the raw html curl response in file fname and parses out an array of quotes. A random index of the quotes array is generated and then the quote at that index is passed to showQuote(). The bg parameter is simply passed through to showQuote() and determines whether or not the quote should be displayed with bg formatting.

`int getQuote(char* movieName, char* movied, int bg)`

Function that makes the curl call to the imdb url. The url is built using the movied parameter. The bg parameter is passed through to Search_in_File(). movieName is passed through as well for display in showQuote().

`char* welcomePage()`

Function that is run at the start of the program. Displays a title header and the mode options. Prompts the user to enter a movie Returns the title of a movie entered by user.

```
void getMovieList(struct movieList* ml)
```

Populates the movieList with the lines contained in the file movie_list so that there is a list of random movies to fetch a quote from if the user chooses. Modifies the memory address pointed to by ml. Calls init_movieList().

```
void init_movieList(struct movieList* ml)
```

Function that initializes the movieList by mallocing space for array ml.movies and sets ml.size and ml.cap.

```
void getRandomQuote(struct movieList movieList)
```

Randomly generates a number in the range [0-movieList.cap-1] and selects the movie title at that index of movieList.movies. Then proceeds fetch a quote in the same manner as if the movie title were input by the user. Prompts the user to approve recursive call to generate another random quote.

```
void showQuote(char* quote, char* title, int bg)
```

Function to display the quote parameter. Also prints the title of the movie that the quote came from. There is some formatting done to keep the text from over running a line. Extra newlines are output if bg parameter is set to 1.

```
char* getMoreQuotes(char* movieName, char* movied, int new)
```

Function that facilitates continued quote searching without needing to restart the program. New determines whether the user will input the title of a new movie. If it is set then the movied must be found by calling returnId(). Ultimately calls getQuote() with the appropriate movieName and movied.

```
do_bg(struct movieList* movies)
```

Function that facilitates running the program in background mode. This function will continually generate a random movie from movies and make the call to getMoreQuotes(). The function contains an infinite while loop as well as a call to sleep(15). This allows the function to output a quote every 15 seconds. Since the program is run in the background it will continue indefinitely, call the makefile target finish to kill this process.

```
int main
```

The main function starts out by determining whether the program has been run in background mode or not. If so then the movieList is constructed and do_bg is called. Otherwise the main function will call the welcomePage to determine the mode and then allow the user to continually generate quotes in the mode of their choosing.

omdb.c

```
char* toHexSpace(const char *s)
```

This Method is used to convert a string with spaces to a string with %20 so that the string may be used in a URL

- @param const char *s
- @return char* b

```
char* returnId(char* title)
```

This method is used to call the OMDB api and parse the return string to return the IMDB id for a specific movie title. This title can then be passed to a url for gaining information on the movie.

- @param char* title
- @return char* idFinal

Development Log

- Using CURL turned out to be a problem with libcurl
 - To fix this we were able to use CURL through a c system call outputting it to a file. Then we were able to parse that file and get the info we needed.

After that we remove the file to clean up.

- Using libjson0 did not work
 - To fix this we simply parsed the OMDB api through CURL as described above. We then parsed out just the IMDB id
- Sending movie titles to the OMDB would not work with a space
 - To counter this we had to create a method toHexSpace() that changed all occurrences of a space in a string to %20. The hex value. This would work when sending the title to the OMDB api