Firstly, a rudimentary text editor is implemented using doubly linked list which is capable of moving up and down the file, and editing the text in its own way. The implementation is done with the help of linked list and structures called editor highlight, editorinit, etc.

The editor works in the following manner, it takes the filename from the command line arguments and then opens it using fopen and gives options to either read, edit and exit.

If r is pressed then the given file is read and the contents of the file is displayed on the terminal. After the content is displayed, there are other options to either move up or down, edit a particular line. To save the edit made the user needs to press s.

All the commands need to be given in small letters only as capital letters are not supported by the code. To implement the file locking, I have used flock with the given two flags:
LOCK_EX and LOCK_NB.

LOCK_EX places an exclusive lock. Only one process may hold an exclusive lock for a given file at a given time.
LOCK_NB is specified so that the EWOULDBLOCK can be handled.

In the main function, the command line argument argv[1] is the file name. The file is opened in both read and write mode using the O_RDWR mode. It is opened using open system call and the returned value is used to apply lock, if the file is not in use then the lock is applied easily and the user can execute the functions. However, if the lock is already in use then only a warning is shown and this is the error value that is handled as the file can be locked only once.

However, even after the warning the user can access the file. The race conditions are not taken into account as if the files are open at two different spaces then the race condition might occur which will require some more code.

For the implementation of the text editor, I have taken help of GeeksforGeeks and referred to the following links:
https://github.com/kyletolle/texor
https://www.sourcecodesolutions.in/2010/09/10implementation-of-simple-text-editor.html