SPECIFICATION TO BE IMPLEMENTED

Implement software that receives **pathnames** as input via argv[].
associated **with N files**, with N greater than or equal to 1. **For each of these file generates a thread** (so in total N new threads will be generated competitors).
Subsequently the main-thread will acquire strings from standard input into an indefinite loop, **and each of the N child threads will have to write every string acquired by the main-thread in the file associated with it**.
The application will have to manage the SIGINT signal (or CTRL_C_EVENT in the case WinAPI) such that when any of the application threads is hit by it will have to print all the strings already in the terminal entered by standard-input and stored in the destination files.

Example. **>> ./a.out file1 file2 file3 file4 ...**

Semaphore structure for thread synchronization.

Using a semaphore array to manage N threads. Assume, for simplicity, N = 5.
The value of N can be any integer.

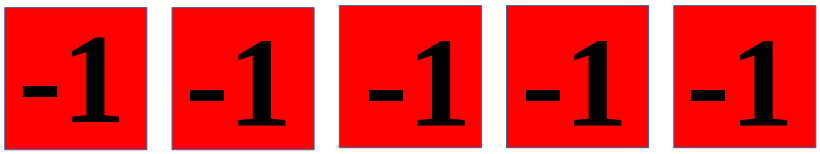
The main thread is managed with a single token dispenser.
N.B: The threads all perform the same operation.

The main threads takes all N available tokens from its dispenser.

Now it acquires the value to insert into the file.

Main releases the token for all threads.

When the main thread works, all the threads are blocked and wait for the main to release a token to unblock them.



Each thread writes to its respective file.

And each thread releases a token to main before starting the loop again.

Restart loop.