Parallel File Copy (Async I/O)

Task

Expected to copy a text file to a defined destination path using Asynchronous I/O operations like download manager. You will copy the content of source file (source.txt) to a new destination file (destination.txt).

- You will take 3 input parameters.
  - Source path (- (dash) means same path as executable file)
  - Destination path (- (dash) means same path as executable file)
  - Number of threads
- Source and destination paths can be given as real paths OR as the character (–dash) which means that related txt file will be created under the same path as executable file.
- Define a function to prepare a source file randomly. The content of the source file must be readable. You will create the source file in main thread. (max size of file 100 MB)
- For example; let's say that the user enters 8 threads and the size of the source file as 16 bytes. So each thread must copy 2 bytes for each. In order to be readable, your source file must be created randomly as "aabbccddeeffgghh".
  - 1st thread should copy from source.txt to destination.txt first 2 bytes > aa
  - 2nd thread should copy from source.txt to destination.txt second 2 bytes > bb
  - …
  - 8th thread should copy from source.txt to destination.txt last 2 bytes > hh
- Create threads (user will be able to define the number of threads from 1 to 10) for asynchronous I/O copy process (read + write).
- Finally, the source.txt file is the exact same content with the destination.txt file.
  - For testing this, you should apply MD5 checksum to verify. MD5 (Message Digest 5) sums can be used as a checksum to verify files or strings in a Linux file system.
    - https://www.tecmint.com/generate-verify-check-files-md5-checksum-linux/