CME 3205 Operating Systems

Assignment #3 (2018-2019 Fall)

Parallel File Copy (Async I/O)

Due date : 23 December 2018, 11:59 pm (No extension, please do not request!!)

Task

In this assignment, you are 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/
- An Example to command-line interface
    - ./run.out - - 8 (Put a space between each parameter.)

CME 3205 Operating Systems

Assignment #3

General Requirements

- For this assignment you will work individual.
- The POSIX AIO and pthread libraries will be used in C prog. lang..
- We compile your code on Debian 8 OS with this code:
    - gcc -lrt -c your_code_file.c
    - gcc your_code_file.o -lrt -o your_code_file.out

Submission

Submission will be via Classroom.

- Name your **code file** as: **StudentNumber.c** (do not use another naming convention.) If you don't follow the naming rules, a penalty applies **(15 pts)**

Grading

- Documenting the code and coding style (proper indentation, describe critical functions) (10 pts)
- User-friendly display (5 pts)
- Command-line interface (10 pts)
- Make source file (15 pts)
- Implementation of Asynchronous I/O (aio_read,aio_write) (35 pts)
- Implementation of thread structure (25 pts)
- **BONUS!** Dynamically display of process percentage of each thread (20 pts)

Honesty

Your submissions will be scanned among each other as well as the Internet repository. Any assignments that are over the similarity threshold of a system for Detecting Software Similarity will get 0. We strongly encourage you not to submit your assignment rather than a dishonest submission.

For Questions

For any questions about the assignment please use Classroom systems comments under Assignment announcement. Before asking your question, please check carefully previous questions and answers, where similar questions were already asked by someone else already answered.

No private questions via email will be answered!!! We will try to answer any of your questions as soon as possible, except the ones "Hocam my code does not work, can you fix it" or "I have implemented it but it does not work, can you look at it". Debuggers are far more suitable options.

Good luck !!!