CSE 341 SYSTEM PROGRAMMİNG HW1 REPORT

In this homework we're asked to execute 4 processes. First two processes and last two process are the same program's processes. ProcessA is the first two processes responsible from reading input files and producing outputfile which is the file will be the input of processB.

In processA, we need to consider reading different input files but writing to common output file. Also this outputfile can be accesed by processB. Because of that reasons, we must use lock mechanism. I created flock structs for lock mechanism. One for write and one for the read lock. This locks used to lock output file since we don't need to lock input files, there is only one user and only reading. After that I created some char buffers to store reading data's to put them after a proper place. I used getopt() to get commandline arguments and store them into some places. If any nonacceptable argument appears, program will closed with INFO message. After that I started to read input files with size of 32 byets. This is a while loop and everytime tries to read 32 byte. If readed byte number is not equal to 32, loop will be terminated. After reading input file, I put readed 32 byte into a buffer array to process data. I processed 32 byte data in 2 byte of chunks, I send this two byte chunk into a function called convertComplex. This function converts 2 byte chunk into complex number and save it to given char pointer place. After that, I put this complex number on a line buffer to write output file after all. With place of number, I putted commas between numbers and '\n' at the end of line. I have one line complex number, now. We can write it to output file but first we must place for writing line. A available place for writing output file is end of file or middle empty lines. For this purpose we must lock output file for reading. First, I checked is there any lock on file. After that when there is no lock, I locked file for reading. I readed all outputFile until finding a line that starts with '\n' char or end of file. After finding place, to avoid overwriting, I created a temp file that hold from cursor point to end of file. I copied all that data to temporary file. After this work done, I get back the cursor to its original point and I locked outputFile for writing. Now, I can write to outputFile. First, I writing converted line and then I writed rest of file from temporary file.i After that I unlocked file, closed temporary file and sleeped the process with given time. When I open files, I used O_SYNC flag to avoid buffering. Also, every temp file must be unique, to do that I opened temporary file with O_TRUNC flag to overwrite. I checked all syscalls and if there is error, I writed error with perror function.

I do not do processB part.