Submit response?

Your username (**hermann@pdx.edu**) and responses will be recorded when you submit this form.

SWITCH ACCOUNT SUBMIT

Assignment 03: filechecker

Work alone or with a partner. If you work with a partner, then indicate their name below AND make sure that your partner also submits this form so that they get credit.

The name, username and photo associated with your Google account will be recorded when you upload files and submit this form. Not **hermann@pdx.edu**? Switch account

* Required

Your Name *

Hermann Yepdjio

Partner's Name (if you worked with a partner).

N/A

filechecker - program to check that files are correctly sorted

create a new C language program called "filechecker.c"
filechecker should take multiple command line arguments
D - directory containing unsorted files. filechecker should assume that D has a sub-directory called "sorted" that contains a sorted version of each file in D.
T - number of checker threads

the program should create T threads and then each of those threads independently checks a subset of the files in D/sorted to make sure that each file is a correctly sorted version of the corresponding file in D. it should assume that the file D/unsorted_<id>.bin has a sorted version called D/sorted/sorted_<id>.bin

the program should exit with 0 (success) if it verifies that all files are properly sorted, and exit with -1 and an appropriate error message if it finds that any sorted_<id>.bin file is not a proper sort of the corresponding unsorted_<id>.bin file.

implementation detail: all file I/O must be done with memory mapped files. The memory maps should be implemented with the mmap(), munmap() and msync() system calls. read the man pages for mmap, munmap and msync to learn more about these important, useful system calls.

Is your code completed and working properly? *

yes

How did you test the filechecker program? i.e., what cases did you try? *

I tried 10 files of 10 integers, 100 files of 100 integers and 1000 files of 10000 integers. I tried t = 1, 2, 10, 100 for the number of threads

Upload your filechecker. c source code. *

filechecker.c ✕

Writing I/O code with mmap() is different from writing I/O code with read() and write(). Comment here about your experiences and preferences with writing I/O code in each style (read/write vs. mmap). *

mmap is so much easier to work with as we read the file data as if we were reading an array but we need to know the size of the file we are reading so that we can figure out when to stop reading. Read/write is not complicated either, but I think mmap is more straight forward

Submit

Never submit passwords through Google Forms.

This form was created inside of Portland State University. Report Abuse