Quiz - File System Performance

Total points 20/20

Take the quiz solo, but feel free to consult a partner student, the book, the videos or other resources if needed. Re-take quiz if your score is less than 80% or if you just want some more practice.

The respondent's email (**faiyaz@pdx.edu**) was recorded on submission of this form.

Which of the following are good reasons/scenarios for which we gain an advantage by using memory mapped file I/O as opposed to doing I/O with calls to read() and write() *

	Yes, this is a good scenario for memory mapped I/O	No, memory mapped I/O does not give a great advantage in this scenario	Score	
many repeated reads and writes of file data	◉	○	1/1	✓
when you anticipate only needing a small number of reads and writes of file data	○	◉	1/1	✓
when a large proportion of your I/O operations are to/from random file locations	◉	○	1/1	✓
when you only plan to read/write a tiny part of a large file	○	◉	1/1	✓
when a single file is shared read-only by many processes	◉	○	1/1	✓

✓ When memory mapping a file, the OS must reserve physical memory space equal to the size of the specified region of the file. *5/5

- ○ True
- ◉ False ✓

✓ An "anonymous" memory mapped file is one in which there is no actual file on the file system, just a region in memory. When the process stores data into the mapped memory region no data is actually written to any file. Why might this be useful? *5/5

- ○ this is useless. there is no use for this feature.
- ◉ if two or more processes map the same "anonymous file" then they can can communicate with each other using memory writes and reads ✓

✓ The concept of a Log-Structured file system is not particularly important for SSDs *5/5

- ◉ True ✓
- ○ False

Feedback

True. With SSDs the OS does not really control the location of the block on the persistent storage device, so all its efforts to optimize/change the locations of the data blocks are ignored.

However, we still teach about it because (1) HDDs are still prevalent and (2) the LFS algorithm greatly influenced the design of SSD flash controllers.