CPSC 3220 – Quiz #11-12 April 16, 2018 Name: _______________

1. (2 points) In Linux, suppose a process successfully opens an existing file that has a single hard link to it, but while the process is reading that file, another process unlinks that file? What happens to subsequent reads by the first process? Do they succeed? Do they fail?

Subsequent reads should still succeed as the file is open, and shouldn't be a problem regardless of what is going on in memory.

2 (2 points) Consider a text editor that saves a file whenever you click a save button. Suppose that when you press the button, the editor simply
(1) animates the button “down” event (e.g., by coloring the button grey),
(2) uses the write() system call to write your text to your file, and then
(3) animates the button “up” event (e.g., by coloring the button white).
What bad thing could happen if a user edits a file, saves it, and then turns off her machine by flipping the power switch (rather than shutting the machine down cleanly)?

A file can be lost, as the original file can be deleted and a new file is moved to the original file's place, but if your computer crashes at a bad time, your system could have no copies of the document left at all.

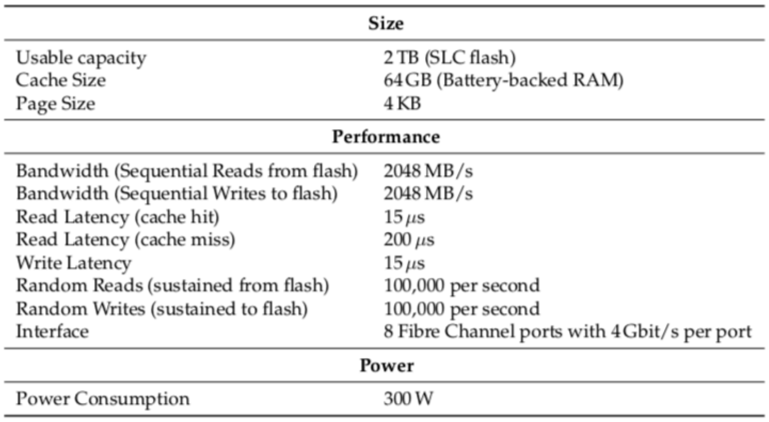
3. (2 points) Discussion. Some high-end disks in the 1980s had multiple disk arm assemblies per disk enclosure in order to allow them to achieve higher performance. Today, high-performance server disks have a single arm assembly per disk enclosure. Why do you think disks so seldom have multiple disk arm assemblies today?

Multiple disk arms would require more physical hardware and more complexity to reads, which will have little increase in read times, therefore the simpler hardware is the best fix to the problem.

Continued on Back

4. (2 points) A disk may have multiple surfaces, arms, and heads, but when you issue a read or write, only one head is active at a time. It seems like one could greatly increase disk bandwidth for large requests by reading or writing with all of the heads at the same time. Given the physical characteristics of disks, can you figure out why no one does this?

Multiple reads and writes with multiple arms could cause thrashing of reads as the complexity would become too much for the reads or writes to complete fast enough, just as the cost would be too great as well.



Size	
Usable capacity	2 TB (SLC flash)
Cache Size	64 GB (Battery-backed RAM)
Page Size	4 KB
Performance	
Bandwidth (Sequential Reads from flash)	2048 MB/s
Bandwidth (Sequential Writes to flash)	2048 MB/s
Read Latency (cache hit)	15 μs
Read Latency (cache miss)	200 μs
Write Latency	15 μs
Random Reads (sustained from flash)	100,000 per second
Random Writes (sustained to flash)	100,000 per second
Interface	8 Fibre Channel ports with 4 Gbit/s per port
Power	
Power Consumption	300 W

5. (2 points) Suppose you have a flash drive such as the one described in the Figure above and you have a workload consisting of 10,000 4 KB reads to pages randomly scattered across the drive. Assuming that you wait for request i to finish before you issue request i + 1, how long will these 10,000 requests take (total)?

10,000 reads * (1 second / 100,000 reads) = 0.1 seconds = 100 ms

6. (2 points) Suppose you have a flash drive such as the one described in Figure above and you have a workload consisting of 10,000 4 KB reads to pages randomly scattered across the drive. Assuming that you issue requests concurrently, using many threads, how long will these 10,000 requests take (total)?

The issuing of requests doesn't speed up the read time, so still 100 ms