

Question 1

None of the disk-scheduling disciplines except FCFS prevents starvation.

- a. This is true because every other disk-scheduling discipline does not take into account how old a request is. Thus, it is always possible to prevent a given request by inserting a ‘higher-priority’ request.
- b. One possible way to modify any algorithm to prevent starvation is to have them take into account the age / time since the request was made, and grant higher priority to those that are very old.
- c. Fairness is important for time-sharing systems as the OS does not always know what processes have what priority, or what the user exactly needs done immediately / in a timely fashion, so it is important to be fair to all processes. One example of fairness being important is if a user wants to load up some files from an SD card while at the same time download a file from the internet, the OS should be able to both read the SD card and save the file in a reasonable time.
- d.
 - If the computer is about to shut down, it might ignore new I/O requests and prioritize critical applications / kernel-level I/O requests.
 - If a process has a very low priority (eg. a background task that takes 10+ hours to do), then the OS might ignore it in favor of more important tasks.
 - If a user is actively using a process, the OS might prioritize that process over others.