

BIRZEIT UNIVERSITY

Computer Systems Engineering Department

ENCS 339 Operating Systems

Second Semester, 2020-2021 HW#4 Mass Storage. Due Date: Monday, January 11th , 2021

**Question 1**: Suppose that a disk drive has 2000 cylinders, numbered 0 to 1999. The drive is currently serving a request at cylinder 150, and the previous request met was at cylinder 140. The queue of pending requests, in FIFO order, is

86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk-scheduling algorithms?

FCFS

SSTF

LOOK

C-SCAN

LCFS

Longest seek time first: LSTF:

Which of the disk-scheduling disciplines mentioned above can cause starvation and which do not. Explain yoir answer. Describe a way to modify algorithms with starvation to ensure absence of starvation.

**Question 2**: Consider a RAID Level 5 organization comprising nine disks, with the parity for sets of eight blocks on four disks stored on the fifth disk. How many blocks are accessed in order to perform the following?

 a.A write of one block of data

b.A write of seven continuous blocks of data.

**Question 3**: A UNIX filesystem has 2-­‐KB blocks and 4-­‐byte disk addresses.

Each i-­‐node contains 10 direct entries, one single-­indirect entry and one double-­‐indirect entry and 1 triple indirect entry..

1. What is the maximum file size?
2. If half of all files are exactly 2.5-­‐KB and the other half of all files are exactly 4-­‐KB, what fraction of disk space would be wasted?
3. Based on the same condition as in b), does it help to reduce the fraction of wasted disk space if we change the block size to 1-­‐KB? Justify your answer.

Consider a system where free space is kept in a free-space list.

1. Suppose that the pointer to the free-space list is lost. Can the system reconstruct the free-space list? Explain your answer.
2. Consider a file system similar to the one used by UNIX with indexed allocation. How many disk I/O operations might be required to read the contents of a small local file at /a/b/c? Assume that none of the disk blocks is currently being cached.

**Question 4**: Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory. For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions in a table form:

1. How is the logical-to-physical address mapping accomplished in this system? (For the indexed allocation, assume that a file is always less than 512 blocks long.)
2. If we are currently at logical block 4 (the last logical block accessed was block 10) and want to access logical block 25, how many physical blocks must be read from the disk?

Good Luck