

Homework 4 for CS153

1. Consider that requests to read the following set of logical block numbers are enqueued to be serviced from a disk that has 100 logical blocks laid out sequentially from block 0 to block 99.

{1, 22, 14, 72, 86, 32, 11, 66, 45, 80}

Assume that the seek time in moving the disk arm head from logical block i to block j is proportional to $|i - j|$. Given that the arm head is currently positioned at block 75 and is in the midst of moving in the direction towards block 0, what is the sequence in which the enqueued blocks will be read with the i) SSTF, ii) SCAN, and iii) C-SCAN algorithms (picks up requests on way down only)?

2. Consider a UNIX-style inode with 10 direct pointer, one single-indirect pointer, and one double-indirect pointer only. Assume that the block size is 8K bytes, and the size of a pointer is 4 bytes.
 - a. What is the largest file size that can be indexed in this system?
 - b. How many blocks (including index blocks) are needed to address a file of size 100 bytes, 10K bytes, 10M bytes, and 4G bytes?
3. We have an empty file system that contains only an empty root directly. Draw a timeline similar to Figure 40.3 and Figure 40.4 in <http://pages.cs.wisc.edu/~remzi/OSTEP/file-implementation.pdf> for each command below.
 - a. Create a subdirectory "bar": `mkdir bar`
 - b. Write "hello" into a new file called "foo" in the subdirectory "bar": `echo hello > /bar/foo`
 - c. Read the file: `cat /bar/foo`
 - d. Delete the file: `rm /bar/foo`