

**CS323 – Exercices**  
**Week 8**  
*9 May 2019*

**1) Disk scheduling**

Consider a disk drive which has 5,000 cylinders (indexed 0 to 4,999). The current request is for cylinder **143** and the previous request was for cylinder **125**.

The queue of pending requests, in FIFO order, is:  
**86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130**

Draw the head movement to process the entire queue and count the total using:

- a) FCFS
- b) SSTF
- c) SCAN
- d) C-SCAN
- e) C-LOOK

**2) Read-ahead and caching using fadvise**

fadvise(2) allows an application to tell the kernel how it expects to use a file handle, so that the kernel can choose appropriate read-ahead and caching techniques for access to the corresponding file. Below are some values can be supplied as advice parameter:

- FADV\_RANDOM : Expect accesses in random order.
- FADV\_SEQUENTIAL : Expect accesses in sequential order.
- FADV\_WILLNEED : Expect access in the near future.
- FADV\_DONTNEED : Do not expect access in the near future.

For each of the above, give at least one example of an application that will benefit from setting the advice parameter to this value.

**3) Direct I/O**

Since Linux 2.4.10, the open(2) system call accepts the O\_DIRECT flag<sup>1</sup>, which allows an application to perform Direct I/O. When Direct I/O is enabled for a file, data is transferred directly from disk to user-space buffers, bypassing the use of the file buffer cache.

- a) What are the performance implications if the file is opened using Direct I/O, in particular with respect to read-ahead and caching ?
- b) Can you think of an application that could benefit from Direct I/O ?

---

<sup>1</sup> The introduction of O\_DIRECT was very controversial. See for example: [http://yarchive.net/comp/linux/o\\_direct.html](http://yarchive.net/comp/linux/o_direct.html)