File-System Management

- File management is one of the most visible components of an operating system. Computers can store information on several different types of physical media. Magnetic disk, optical disk, and magnetic tape are the most common. Each of these media has its own characteristics and physical organization. Each medium is controlled by a device, such as a disk drive or tape drive, that also has its own unique characteristics. These properties include access speed, capacity, data-transfer rate, and access method (sequential or random).
- A file is a collection of related information defined by its creator. Commonly, files represent programs (both source and object forms) and data. Data files may be numeric, alphabetic, alphanumeric, or binary. Files may be free-form (for example, text files), or they may be formatted rigidly (for example, fixed fields). Clearly, the concept of a file is an extremely general one.
- The operating system implements the abstract concept of a file by managing mass-storage media, such as tapes and disks, and the devices that control them.
- The operating system is responsible for the following activities in connection with file management:
    - Creating and deleting files
    - Creating and deleting directories to organize files
    - Supporting primitives for manipulating files and directories
    - Mapping files onto secondary storage
    - Backing up files on stable (nonvolatile) storage media

Mass-Storage Management

- As we have already seen, because main memory is too small to accommodate all data and programs, and because the data that it holds are lost when power is lost, the computer system must provide secondary storage to back up main memory. Most modern computer systems use disks as the principal on-line storage medium for both programs and data. Most programs—including compilers, assemblers, word processors, editors, and formatters—are stored on a disk until loaded into memory. They then use the disk as both the source and destination of their processing. Hence, the proper management of disk storage is of central importance to a computer system. The operating system is responsible for the following activities in connection with disk management:

  - Free-space management
  - Storage allocation
  - Disk scheduling
- Because secondary storage is used frequently, it must be used efficiently. The entire speed of operation of a computer may hinge on the speeds of the disk subsystem and the algorithms that manipulate that subsystem.
- There are, however, many uses for storage that is slower and lower in cost (and sometimes of higher capacity) than secondary storage. Backups of disk data, storage of seldom-used data, and long-term archival storage are some examples. Magnetic tape drives and their tapes and CD and DVD drives and platters are typical tertiary storage devices. The media (tapes and optical platters) vary between WORM (write-once, read-many-times) and RW (read– write) formats.
- Tertiary storage is not crucial to system performance, but it still must be managed. Some operating systems take on this task, while others leave tertiary-storage management to application programs. Some of the functions that operating systems can provide include mounting and unmounting media in devices, allocating and freeing the devices for exclusive use by processes, and migrating data from secondary to tertiary storage.

Disk scheduling

One of the responsibilities of the operating system is to use the hardware efficiently. For the disk drives, efficiency means having fast access time and large disk bandwidth.

The disk **bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer. We can improve both the access time and the bandwidth by managing the order in which disk I/O requests are serviced.

Whenever a process needs I/O to or from the disk, it issues a system call to the operating system. The request specifies several pieces of information:

- Whether this operation is input or output
- What is the disk address for the transfer
- What is the memory address for the transfer
- What is the number of sectors to be transferred
- If the desired disk drive and controller are available, the request can be serviced immediately. If the drive or controller is busy, any new requests for service will be placed in the queue of pending requests for that drive. Thus, when one request is completed, the operating system chooses which pending request to service next. How does the operating system make this choice? Any one of several disk-scheduling algorithms can be used.

FCFS (First Come First Serve)

The simplest form of disk scheduling is, of course, the first-come, first-served (FCFS) algorithm. In FCFS, the requests are addressed in the order they arrive in the disk queue. This algorithm is intrinsically fair, but it generally does not provide less average seek time.

Advantages:

- Easy to understand easy to use
- Every request gets a fair chance
- no starvation (may suffer from convoy effect)

Disadvantages:

- Does not try to optimize seek time (extra seek movements)
- May not provide the best performance based on different parameters

Q If the disk head is located initially at 32, find the number of disk moves required with FCFS if the disk queue of I/O blocks requests are 98, 37, 14, 124, 65, 67. **(NET-DEC-2012)**

(A) 239 **(B)** 310 **(C)** 321 **(D)** 325

Answer: (C)

Q Assuming that the disk head is located initially at 32, find the number of disk moves required with FCFS if the disk queue of I/O block requests are 98, 37, 14, 124, 65, 67: **(NET-JUNE-2013)**

(A) 310 **(B)** 324 **(C)** 320 **(D)** 321

Answer: (D)

Q If the Disk head is located initially at track 32, find the number of disk moves required with FCFS scheduling criteria if the disk queue of I/O blocks requests are: 98, 37, 14, 124, 65, 67 **(NET-JULY-2016)**

a) 320 **b)** 322 **c)** 321 **d)** 319

Answer: (C)

SSTF Scheduling

It seems reasonable to service all the requests close to the current head position before moving the head far to service other REQUESTS. This assumption is the basis for the SSTF algorithm.

In SSTF (Shortest Seek Time First), the request nearest to the disk arm will get executed first i.e. requests having shortest seek time are executed first.

Although the SSTF algorithm is a substantial improvement over the FCFS algorithm, it is not optimal. In the example, we can do better by moving the head from 53 to 37, even though the latter is not closest, and then to 14, before turning around to service 65, 67, 98, 122, 124, and 183. This strategy reduces the total head movement to 208 cylinders.

Advantages:

- Seek movements decreases
- Average Response Time decreases
- Throughput increases

Disadvantages:

- Overhead to calculate the closest request.
- Can cause Starvation for a request which is far from the current location of the header
- High variance of response time as SSTF favours only some requests

SSTF scheduling is essentially a form of shortest-job-first (SJF) scheduling; and like SJF scheduling, it may cause starvation of some requests.

Q Consider a disk system with 100 cylinders. The requests to access the cylinders occur in following sequence 4, 34, 10, 7, 19, 73, 2, 15, 6, 20. Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes 1ms to move from one cylinder to adjacent one and shortest seek time first policy is used? (GATE-2009) (1 Marks)

(A) 95 ms **(B)** 119 ms **(C)** 233 ms **(D)** 276 ms

Answer: (B)

Q Suppose a disk has 201 cylinders, numbered from 0 to 200. At some time, the disk arm is at cylinder 100, and there is a queue of disk access requests for cylinders 30, 85, 90, 100, 105, 110, 135 and 145. If Shortest-Seek Time First (SSTF) is being used for scheduling the disk access, the request for cylinder 90 is serviced after servicing ____________ number of requests. **(GATE-2014) (1 Marks)**
Answer: 3

Q Consider an imaginary disk with 40 cylinders. A request come to read a block on cylinder 11. While the seek to cylinder 11 is in progress, new requests come in for cylinders 1, 36, 16, 34, 9 and 12 in that order. The number of arm motions using shortest seek first algorithm is **(NET-DEC-2014)**

(A) 111 **(B)** 112 **(C)** 60 **(D)** 61

Answer: (D)

Q Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 37, 122, 14, 124, 65, 67. Suppose SSTF disk scheduling algorithm implemented to meet the requests then the total number of head movements are __________ if the disk head is initially at 53. **(NET-NOV-2017)**

a) 224 **b)** 248 **c)** 236 **d)** 240

Answer: (C)

Q A disk drive has 100 cylinders, numbered 0 to 99. Disk requests come to the disk driver for cylinders 12, 26, 24, 4, 42, 8 and 50 in that order. The driver is currently serving a request at cylinder 24. A seek takes 6 msec per cylinder moved. How much seek time is needed for shortest seek time first (SSTF) algorithm? **(NET-JUNE-2015)**

a) 0.984 sec **b)** 0.396 sec **c)** 0.738 sec **d)** 0.42 sec

Answer: (D)

SCAN

The disk arm starts at one end of the disk and moves towards the other end, servicing requests as it reaches each track, until it gets to the other end of the disk. At the other end, the direction of head movement is reversed, and servicing continues. The head continuously scans back and forth across the disk. The SCAN algorithm is sometimes called the Elevator algorithm, since the disk arm behaves just like an elevator in a building, first servicing all the requests going up and then reversing to service requests the other way.

If a request arrives in the queue just in front of the head, it will be serviced almost immediately; a request arriving just behind the head will have to wait until the arm moves to the end of the disk, reverses direction, and comes back.

Assuming a uniform distribution of requests for cylinders, consider the density of requests when the head reaches one end and reverses direction. At this point, relatively few requests are immediately in front of the head since these cylinders have recently been serviced. The heaviest density of requests is at the other end of the disk These requests have also waited the longest.

Advantages:

- Simple easy to understand and use
- No starvation but more wait for some random process
- Low variance and Average response time

Disadvantages:

- Long waiting time for requests for locations just visited by disk arm.
- Unnecessary move to the end of the disk, even if there is no request.

Q On a disk with 1000 cylinders (0 to 999) find the number of tracks, the disk arm must move to satisfy all the requests in the disk queue. Assume the last request service was at track 345 and the head is moving toward track 0. The queue in FIFO order contains requests for the following tracks: 123, 874, 692, 475, 105, 376 (Assume SCAN algorithm) **(NET-DEC-2012)**

(A) 2013 **(B)** 1219 **(C)** 1967 **(D)** 1507

Answer: (B)

Q Consider a disk queue with I/O requests on the following cylinders in their arriving order: 6, 10, 12, 54, 97, 73, 128, 15, 44, 110, 34, 45 The disk head is assumed to be at cylinder 23 and moving in the direction of decreasing number of cylinders. Total number of cylinders in the disk is 150. The disk head movement using SCAN-scheduling algorithm is: **(NET-JAN-2017)**

a) 172 **b)** 173 **c)** 227 **d)** 228

Answer: (all options are incorrect 151)

Q Suppose the following disk request sequence (track numbers) for a disk with 100 tracks is given: 45, 20, 90, 10, 50, 60, 80, 25, 70. Assume that the initial position of the R/W head is on track 50. The additional distance that will be traversed by the R/W head when the Shortest Seek Time First (SSTF) algorithm is used compared to the SCAN (Elevator) algorithm (assuming that SCAN algorithm moves towards 100 when it starts execution) is __________ tracks **(GATE-2015) (2 Marks)**

Answer: 10

Q Consider a disk queue with request for input/output to block on cylinders 98, 183, 37, 122, 14, 124, 65, 67 in that order. Assume that disk head is initially positioned at cylinder 53 and moving towards cylinder number 0. The total number of head movements using Shortest Seek Time First (SSTF) and SCAN algorithms are respectively **(NET-DEC-2013)**

(A) 236 and 252 cylinders **(B)** 640 and 236 cylinders

(C) 235 and 640 cylinders **(D)** 235 and 252 cylinders

Answer: (all options are incorrect)

C-SCAN Scheduling

Circular-scan is a variant of SCAN designed to provide a more uniform wait time. Like SCAN, C-SCAN moves the head from one end of the disk to the other, servicing requests along the way. When the head reaches the other end, however, it immediately returns to the beginning of the disk without servicing any requests on the return trip.

Advantages:

- Provides more uniform wait time compared to SCAN
- Better response time compared to scan

Disadvantage:

- More seeks movements in order to reach starting position

Q Consider the situation in which the disk read/write head is currently located at track 45 (of tracks 0-255) and moving in the positive direction. Assume that the following track requests have been made in this order: 40, 67, 11, 240, 87. What is the order in which optimized C-SCAN would service these requests and what is the total seek distance?**(GATE-2015) (2 Marks)**

(A) 600 **(B)** 810 **(C)** 505 **(D)** 550

Answer: (D)

LOOK Scheduling

It is similar to the SCAN disk scheduling algorithm except the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus, it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Advantage: -

- Better performance compared to SCAN
- Should be used in case to less load

Disadvantage: -

- Overhead to find the last request
- Should not be used in case of more load.

C LOOK

As LOOK is similar to SCAN algorithm, in similar way, C-LOOK is similar to C-SCAN disk scheduling algorithm. In C-LOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end's last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Advantage: -

- Provides more uniform wait time compared to LOOK
- Better response time compared to LOOK

Disadvantage: -

- Overhead to find the last request and go to initial position is more
- Should not be used in case of more load.

Q Which of the following statements is not true about disk-arm scheduling algorithms? **(NET-JUNE-2014)**

(A) SSTF (shortest seek time first) algorithm increases performance of FCFS.

(B) The number of requests for disk service are not influenced by file allocation method.

(C) Caching the directories and index blocks in main memory can also help in reducing disk arm movements.

(D) SCAN and C-SCAN algorithms are less likely to have a starvation problem.

Answer: (B)

Q In __________ disk scheduling algorithm, the disk head moves from one end to other end of the disk, serving the requests along the way. When the head reaches the other end, it immediately returns to the beginning of the disk without serving any requests on the return trip. **(NET-NOV-2017)**

a) LOOK **b)** SCAN **c)** C-LOOK **d)** C-SCAN

Answer: (D)

Q Suppose the following disk request sequence (track numbers) for a disk with 100 tracks is given: 45, 20, 90, 10, 50, 60, 80, 25, 70. Assume that the initial position of the R/W head is on track 50. The additional distance that will be traversed by the R/W head when the Shortest Seek Time First (SSTF) algorithm is used compared to the SCAN (Elevator) algorithm (assuming that SCAN algorithm moves towards 100 when it starts execution) is __________ tracks(GATE-2015) (2 Marks)

Answer: 10

Q Consider a disk queue with requests for I/O to blocks on cylinders 47, 38, 121, 191, 87, 11, 92, 10. The C-LOOK scheduling algorithm is used. The head is initially at cylinder number 63, moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is ____________. **(GATE-2018) (2 Marks)**

Answer: 346

Conclusion

Selection of a Disk-Scheduling Algorithm

- Given so many disk-scheduling algorithms, how do we choose the best one? SSTF is common and has a natural appeal because it increases performance over FCFS. SCAN and C-SCAN perform better for systems that place a heavy load on the disk, because they are less likely to cause a starvation problem.
- With any scheduling algorithm, however, performance depends heavily on the number and types of requests. For instance, suppose that the queue usually has just one outstanding request. Then, all scheduling algorithms behave the same, because they have only one choice of where to move the disk head: they all behave like FCFS scheduling.
- Requests for disk service can be greatly influenced by the file-allocation method. A program reading a contiguously allocated file will generate several requests that are close together on the disk, resulting in limited head movement. A linked or indexed file, in contrast, may include blocks that are widely scattered on the disk, resulting in greater head movement.
- The location of directories and index blocks is also important. Since every file must be opened to be used, and opening a file requires searching the directory structure, the directories will be accessed frequently. Suppose that a directory entry is on the first cylinder and a file's data are on the final cylinder. In this case, the disk head has to move the entire width of the disk. If the directory entry were on the middle cylinder, the head would have to move only one-half the width. Caching the directories and index blocks in main memory can also help to reduce disk-arm movement, particularly for read requests.
- Because of these complexities, the disk-scheduling algorithm should be written as a separate module of the operating system, so that it can be replaced with a different algorithm if necessary. Either SSTF or LOOK is a reasonable choice for the default algorithm.
- The scheduling algorithms described here consider only the seek distances. For modern disks, the rotational latency can be nearly as large as the average seek time. It is difficult for the operating system to schedule for improved rotational latency, though, because modern disks do not disclose the physical location of logical blocks. Disk manufacturers have been alleviating this problem by implementing disk-scheduling algorithms in the controller hardware built into the disk drive. If the operating system sends a batch of requests to the controller, the controller can queue them and then schedule them to improve both the seek time and the rotational latency.
- If I/O performance were the only consideration, the operating system would gladly turn over the responsibility of disk scheduling to the disk hard- ware. In practice, however, the operating system may have other constraints on the service order for requests. For instance, demand paging may take priority over application I/O, and writes are more urgent than reads if the cache is running out of free pages.

- Also, it may be desirable to guarantee the order of a set of disk writes to make the file system robust in the face of system crashes. Consider what could happen if the operating system allocated a disk page to a file and the application wrote data into that page before the operating system had a chance to flush the file system metadata back to disk. To accommodate such requirements, an operating system may choose to do its own disk scheduling and to spoon-feed the requests to the disk controller, one by one, for some types of I/O.
  - SSTF is common and has a natural appeal because it increases performance over FCFS.
  - SCAN and C-SCAN perform better for systems that place a heavy load on the disk, because they are less likely to cause a starvation problem.
  - With any scheduling algorithm however, performance depends heavily on the number and types of requests.
  - For instance, suppose that the queue usually has just one outstanding request. Then all scheduling algorithms behave the same because they have only one choice of where to move the disk head: they all behave like FCFS scheduling.
  - Requests for disk service can be greatly influenced by the file-allocation method.
  - Because of these complexities, the disk-scheduling algorithm should be written as a separate module of the operating system, so that it can be replaced with a different algorithm if necessary. Either SSTF or LOOK is a reasonable choice for the default algorithm. The scheduling algorithms described here consider only the seek distances

Q Consider an operating system capable of loading and executing a single sequential user process at a time. The disk head scheduling algorithm used is First Come First Served (FCFS). If FCFS is replaced by Shortest Seek Time First (SSTF), claimed by the vendor to give 50% better benchmark results, what is the expected improvement in the I/O performance of user programs? **(GATE-2004) (1 Marks)**

(A) 50% **(B)** 40% **(C)** 25% **(D)** 0%

Answer: (D)

Q Which of the following disk scheduling strategies is likely to give the best throughput? **(GATE-1999) (1 Marks)**

(a) Farthest cylinder next
(b) Nearest cylinder next
(c) First come first served
(d) Elevator algorithm

Answer: (B)

Q Match the following **(NET-DEC-2004)**

(a) Disk scheduling	(1) Round robin
(b) Batch processing	(2) Scan
(c) Time sharing	(3) LIFO
(d) Interrupt processing	(4) FIFO

Codes:

(A)	a-3	b-4	c-2	d-1
(B)	a-4	b-3	c-2	d-1
(C)	a-2	b-4	c-1	d-3
(D)	a-3	b-4	c-1	d-2

Answer: (C)

Q The maximum amount of information that is available in one portion of the disk access arm for a removal disk pack (without further movement of the arm with multiple heads) **(NET-DEC-2011)**

(A) a plate of data
(B) a cylinder of data
(C) a track of data
(D) a block of data

Answer: (B)

Q Consider the input/output (I/O) requests made at different instants of time directed at a hypothetical disk having 200 tracks as given in the following table

Serial No	1	2	3	4	5
Track No	12	85	40	100	75
Time of arrival	65	80	110	100	175

Assume that:

Current head position is at track no. 65 Direction of last movement is towards higher numbered tracks Current clock time is 160 milliseconds Head movement time per track is 1 millisecond. "look" is a variant of "SCAN" disk- arm scheduling algorithm. In this algorithm, if no more I/O requests are left in current direction, the disk head reverses its direction. The seek times in Shortest Seek First (SSF) and "look" disk-arm scheduling algorithms respectively are: **(NET-SEP-2013)**

(A) 144 and 123 milliseconds **(B)** 143 and 123 milliseconds

(C) 149 and 124 milliseconds **(D)** 256 and 186 milliseconds

Answer: (B)

Q An operating system supports a paged virtual memory, using a central processor with a cycle time of one microsecond. It costs an additional one microsecond to access a page other than the current one. Pages have 1000 words, and the paging device is a drum that rotates at 3000 revolutions per minute and transfers one million words per second. Further, one percent of all instructions executed accessed a page other than the current page. The instruction that accessed another page, 80% accessed a page already in memory and when a new page was required, the replaced page was modified 50% of the time. What is the effective access time on this system, assuming that the system is running only one process and the processor is idle during drum transfers? **(NET-AUG-2016)**

A) 30 microseconds **b)** 34 microseconds **c)** 60 microseconds **d)**68 microseconds

Answer: (B)

Q Consider a storage disk with 4 platters (numbered as 0, 1, 2 and 3), 200 cylinders (numbered as 0, 1, …, 199), and 256 sectors per track (numbered as 0, 1, … 255). The following 6 disk requests of the form [sector number, cylinder number, platter number] are received by the disk controller at the same time:

[120, 72, 2], [180, 134, 1], [60, 20, 0], [212, 86, 3], [56, 116, 2], [118, 16, 1]

Currently head is positioned at sector number 100 of cylinder 80, and is moving towards higher cylinder numbers. The average power dissipation in moving the head over 100 cylinders is 20 milliwatts and for reversing the direction of the head movement once is 15 milliwatts. Power dissipation associated with rotational latency and switching of head between different platters is negligible.

The total power consumption in milliwatts to satisfy all of the above disk requests using the Shortest Seek Time First disk scheduling algorithm is _______. **(GATE-2018) (2 Marks)**

Answer: 85

Q For a magnetic disk with concentric circular tracks, the seek latency is not linearly proportional to the seek distance due to **(GATE-2008) (1 Marks)**

a) non-uniform distribution of requests

b) arm starting and stopping inertia

c) higher capacity of tracks on the periphery of the platter

d) use of unfair arm scheduling policies

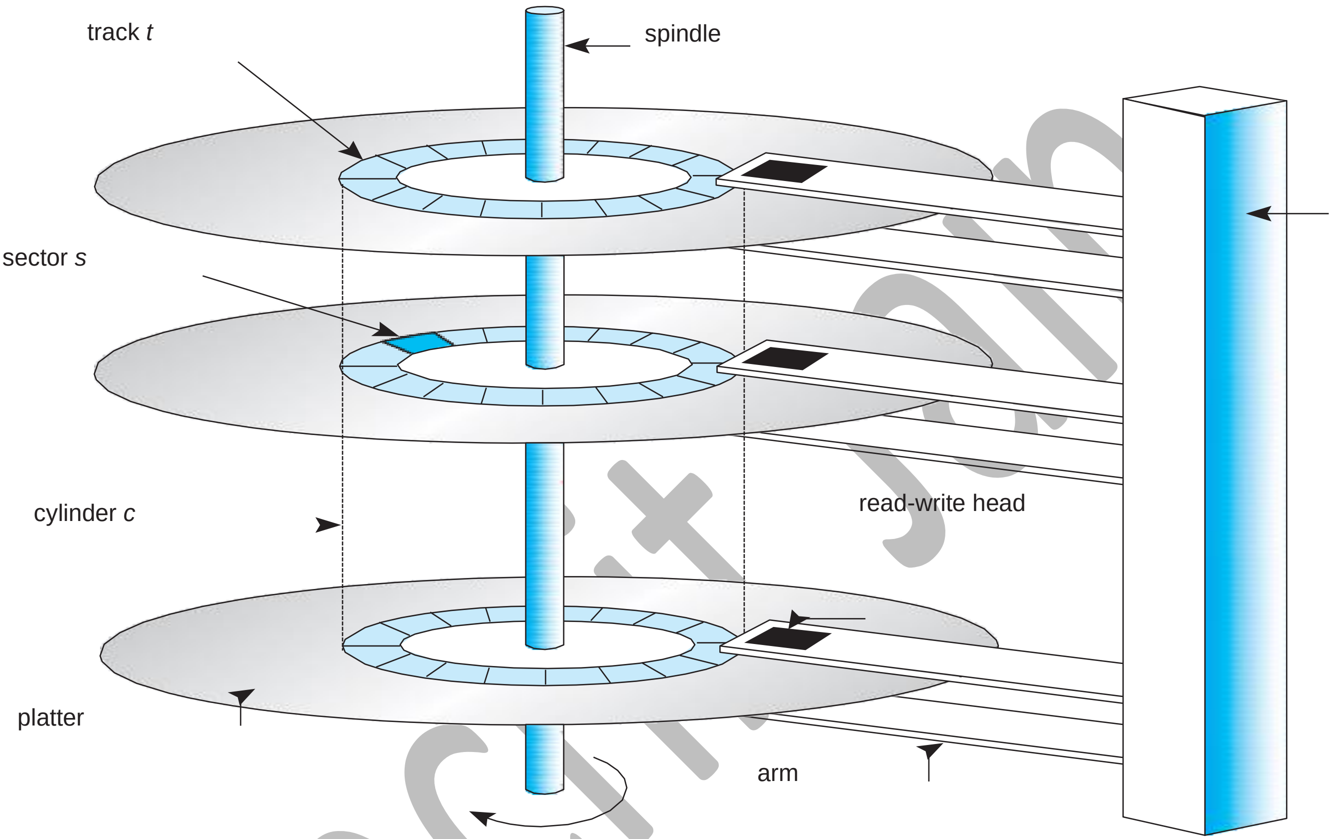
Answer B

Q Consider a hard disk with 16 recording surfaces (0-15) having 16384 cylinders (0-16383) and each cylinder contains 64 sectors (0-63). Data storage capacity in each sector is 512 bytes. Data are organized cylinder-wise and the addressing format is <cylinder no., surface no., sector no.>. A file of size 42797 KB is stored in the disk and the starting disk location of the file is <1200, 9, 40>. What is the cylinder number of the last sector of the file, if it is stored in a contiguous manner? **(GATE-2013) (2 Marks)**

Answer: 1284

Transfer time

- Magnetic disks provide the bulk of secondary storage for modern computer systems. Each disk platter has a flat circular shape, like a CD. Common platter diameters range from 1.8 to 3.5 inches. The two surfaces of a platter are covered with a magnetic material. We store information by recording it magnetically on the platters.



- A read–write head "flies" just above each surface of every platter. The heads are attached to a disk arm that moves all the heads as a unit. The surface of a platter is logically divided into circular tracks, which are subdivided into sectors. The set of tracks that are at one arm position makes up a cylinder. There may be thousands of concentric cylinders in a disk drive, and each track may contain hundreds of sectors.

- When the disk is in use, a drive motor spins it at high speed. Most drives rotate 60 to 250 times per second, specified in terms of rotations per minute **(RPM)**. Common drives spin at 5,400, 7,200, 10,000, and 15,000 RPM.
- Total transfer time has two parts. The **positioning time**, or **random-access time**, consists of two parts: the time necessary to move the disk arm to the desired cylinder, called the **seek time**, and the time necessary for the desired sector to rotate to the disk head, called the **rotational latency**.

Q Consider a disk where there are 512 tracks, each track is capable of holding 128 sector and each sector holds 256 bytes, find the capacity of the track and disk and number of bits required to reach correct track, sector and disk.

Q consider a disk where each sector contains 512 bytes and there are 400 sectors per track and 1000 tracks on the disk. If disk is rotating at speed of 1500 RPM, find the total time required to transfer file of size 1 MB. Suppose seek time is 4ms?

Q Consider a system with 8 sector per track and 512 bytes per sector. Assume that disk rotates at 3000 rpm and average seek time is 15ms standard. Find total time required to transfer a file which requires 8 sectors to be stored.

a) Assume contiguous allocation

b) Assume Non- contiguous allocation

Total Transfer Time = Seek Time + Rotational Latency + Transfer Time

Seek Time: - It is a time taken by Read/Write header to reach the correct track. (Always given in question)

Rotational Latency: - It is the time taken by read/Write header during the wait for the correct sector. In general, it's a random value, so far average analysis, we consider the time taken by disk to complete half rotation.

Transfer Time: - it is the time taken by read/write header either to read or write on a disk. In general, we assume that in 1 complete rotation, header can read/write the either track, so total time will be

(File Size/Track Size) *time taken to complete one revolution.

Q Consider a disk pack with 16 surfaces, 128 tracks per surface and 256 sectors per track. 512 bytes of data are stored in a bit serial manner in a sector. The capacity of the disk pack and the number of bits required to specify a particular sector in the disk are respectively(GATE-2006) (2 Marks)

(A) 256 Mbyte, 19 bits
(B) 256 Mbyte, 28 bits
(C)512 Mbyte, 20 bits
(D) 64 Gbyte, 28 bit

Answer: (A)

Q Consider a disk with 16384 bytes per track having a rotation time of 16 msec and average seek time of 40 msec. What is the time in msec to read a block of 1024 bytes from this disk? **(NET-DEC-2015)**

a) 57 sec
b) 49 sec
c) 48 sec
d) 17 sec

Answer: (B)

Q An application loads 100 libraries at start-up. Loading each library requires exactly one disk access. The seek time of the disk to a random location is given as 10 milli second. Rotational speed of disk is 6000 rpm. If all 100 libraries are loaded from random locations on the disk, how long does it take to load all libraries? (The time to transfer data from the disk block once the head has been positioned at the start of the block may be neglected) (GATE-2011) (2 Marks)

(A) 0.50 s
(B) 1.50 s
(C) 1.25 s
(D) 1.00 s

Answer: (B)

Q Consider a disk pack with a seek time of 4 milliseconds and rotational speed of 10000 rotations per minute (RPM). It has 600 sectors per track and each sector can store 512 bytes of data. Consider a file stored in the disk. The file contains 2000 sectors. Assume that every sector access necessitates a seek, and the average rotational latency for accessing each sector is half of the time for one complete rotation. The total time (in milliseconds) needed to read the entire file is _________. (GATE-2015) (2 Marks)

Answer: 14020

Q Consider a typical disk that rotates at 15000 rotations per minute (RPM) and has a transfer rate of 50×10^6 bytes/sec. If the average seek time of the disk is twice the average rotational delay and the controller's transfer time is 10 times the disk transfer time, the average time (in milliseconds) to read or write a 512 byte sector of the disk is ______________ (GATE-2015) (2 Marks)

Answer: 6.1

Q For a magnetic disk with concentric circular tracks, the seek latency is not linearly proportional to the seek distance due to(GATE-2008) (2 Marks)

(A) non-uniform distribution of requests

(B) arm starting and stopping inertia

(C) higher capacity of tracks on the periphery of the platter

(D) use of unfair arm scheduling policies

Answer: (B)

Q Consider a disk pack with 16 surfaces, 128 tracks per surface and 256 sectors per track. 512 bytes of data are stored in a bit serial manner in a sector. The capacity of the disk pack and the number of bits required to specify a particular sector in the disk are respectively(GATE-2006) (2 Marks)

(A) 256 Mbyte, 19 bits

(B) 256 Mbyte, 28 bits

(C) 512 Mbyte, 20 bits

(D) 64 Gbyte, 28 bit

Answer: (A)

Q Using a larger block size in a fixed block size file system leads to **(GATE-2003) (1 Marks)**

(A) better disk throughput but poorer disk space utilization

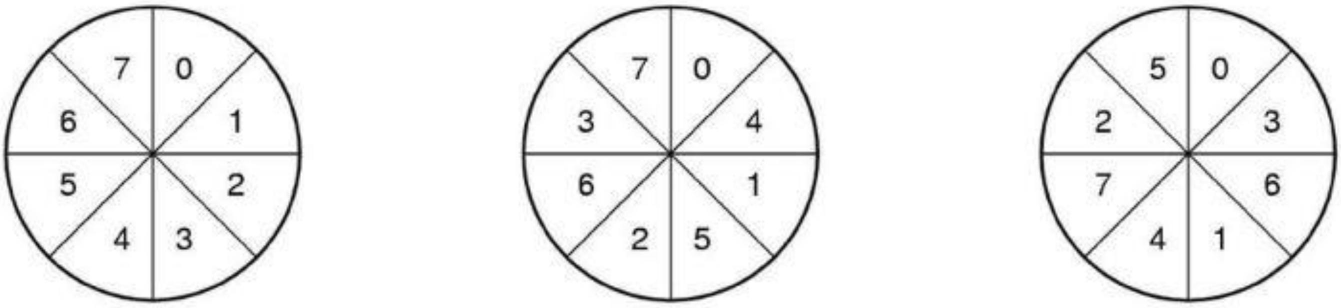
(B) better disk throughput and better disk space utilization

(C) poorer disk throughput but better disk space utilization

(D) poorer disk throughput and poorer disk space utilization

Answer: (A)

Disk interleaving



(a) (b) (c)

- No interleaving
- Single interleaving
- Double interleaving

Single interleaving: - in 2 rotation we read 1 track

Double interleaving: - in 2.75 rotation we read 1 track