IARE '24 DISCORD TEAM

II-II

OS MODULE 4 SOLUTIONS

AD • UJJWAL • ASRITHA • PRANAV

FILE SYSTEM INTERFACE, MASS-STORAGE STRUCTURE

Few Pending

OS MODULE 4

PART A

1. A hard disk has 63 sectors per track, 10 platters each with 2 recording surfaces and 1000 cylinders. The address of a sector is given as a triple ⟨c,h,s⟩, where c is the cylinder number, h is the surface number and s is the sector number. Thus, the 0th sector is addresses as ⟨0,0,0⟩, the 1st sector as ⟨0,0,1⟩, and so on. The address of the 1050th sector is

Address triple = (c, h, s)

c ⟶ cylinder number

h ⟶ surface number

S ⟶ sector number

10 platters (with 2 surfaces)

63 sectors per track

1000 cylinders

1 cylinder = 63 x 20 sectors = 1260

1 surface = 63 sectors to cylinder

1050th Sector ⟶ Address

c = 0 (bcuz 1 cylinder needs 1260 sectors)

$h = \frac{1050}{63} = 16.66 = 16$

S = 1050 mod 63 = 42

1050th sector Address ⟶ <0,16,42>

2. Explain the maximum file size supported by a file system with 16 direct blocks, single, double, and triple indirection. The block size is 512 bytes. Disk block numbers can be stored in 4 bytes.

We have, block size = 512

number of block numbers in an indirection block

$= \text{block size} / 4$

$= 128$

number of blocks for file data in that file object

$= 16 + 128 + 128^2 + 128^3$

Maximum file size:

(direct + single indirect + double indirect + triple indirect) * (blocksize)

$= (16 + 512/4 + (512/4)^2 + (512/4)^3) * (512)$

$= 68853964800$ bytes, ~64 GB

3. Discuss the reasons why the operating system might require accurate information on how blocks are stored on disk. how could operating system improves file system performance with this knowledge

While allocating blocks for a file, the operating system could allocate blocks that are geometrically close by on the disk if it had more information regarding the physical location of the blocks on the disk. In particular, it could allocate a block of data and then allocate the second block of data in the same cylinder but on a different surface at a rotationally optimal place so that the access to the next block could be made with minimal cost.

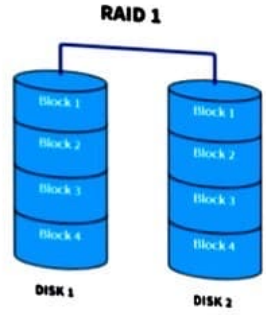
4. Discuss how the OS could maintain a free-space list for a tape-resident file system. Assume that the tape technology is append-only and that it uses EOT marks and locate, space and read position command

- Since this tape technology is append-only, all the free space is at the end of the tape.

- The location of this free space does not need to be stored at all, because the space command can be used to position to the EOT mark.
- The amount of available free space after the EOT mark can be represented by a single number.
- It may be desirable to maintain a second number to represent the amount of space occupied by files that have been logically deleted (but their space has not been reclaimed since the tape is append-only) so that we can decide when it would pay to copy the non-deleted files to a new tape in order to reclaim the old tape for reuse.
- We can store the free and deleted space numbers on disk for easy access. Another copy of these numbers can be stored at the end of the tape as the last data block.
- We can overwrite this last data block when we allocate new storage on the tape.

5. Compare the performance of write operations achieved by a RAID level 5 organization with that achieved by a RAID level 1 organization.

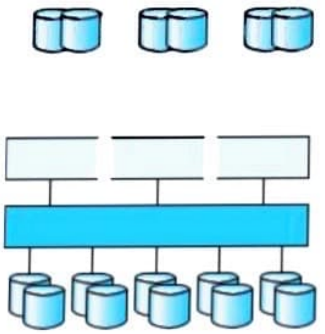
RAID Level 1 organization can perform writes by simply issuing the writes to mirrored data concurrently. RAID Level 5, on the other hand, would require the old contents of the parity block to be read before it is updated based on the new contents of the target block. This results in more overhead for the write operations on a RAID Level 5 system.



6. Is there any way to implement truly stable storage. Explain your answer.

By definition, information residing in the Stable-Storage is never lost. Even if the disk and CPU have some errors, it will never lose any data.

Stable-Storage Implementation :
To achieve such storage, we need to replicate the required information on multiple storage devices with independent failure modes. The writing of an update should be coordinated in such a way that it would not delete all the copies of the state and that, when we are recovering from a failure, we can force all the copies to a consistent and correct value, even if another failure occurs during the recovery. Truly stable storage would never lose data. The fundamental technique for stable storage is to maintain multiple copies of the data, so that if one copy is destroyed, some other copy is still available for use. But for any scheme, we can imagine a large enough disaster that all copies are destroyed.



Multiple Storages in Stable-Storage Implementation

7. What are file protection methods?

Protection mechanisms provide controlled access by limiting the types of file access that can be made. Access is permitted or denied depending on several factors, one of which is the type of access requested. Several different types of operations may be controlled:

- **Read:** Read from the file.
- **Write:** Write or rewrite the file.
- **Execute:** Load the file into memory and execute it.
- **Append:** Write new information at the end of the file.
- **Delete:** Delete the file and free its space for possible reuse.
- **List:** List the name and attributes of the file.

Other Protection Approaches:

The access to any system is also controlled by the password. If the use of password is random and it is changed often, this may result in limiting the effective access to a file.

The use of passwords has a few disadvantages:

- The number of passwords is very large so it is difficult to remember the large passwords.
- If one password is used for all the files, then once it is discovered, all files are accessible; protection is on all-or-none basis.

8. Explain different types of files.

The types of files recognized by the system are either regular, directory, or special. However, the operating system uses many variations of these basic types.

The following basic types of files exist:

Item	Description
regular	Stores data (text, binary, and executable)
directory	Contains information used to access other files
special	Defines a FIFO (first-in, first-out) pipe file or a physical device

BIOS	Contains information related to Basic Input Output System

Regular file: It stores data of text, binary information these are the most common files. It is also known as ordinary files.

This is of two types:

(i) Text file: It contains information that is readable by the user. We can display and print these files.

(ii) Binary files: It contains information that is readable by the computer. These files may be executable files that instruct the system to accomplish a job.

Directory files: It contains information about a system that needs to access all types of files, but they do not contain the actual file data.

It occupies less space than a regular file.

Special files: These are the temporary files created by process.

These are of three types:

(i) FIFO

(ii) Block

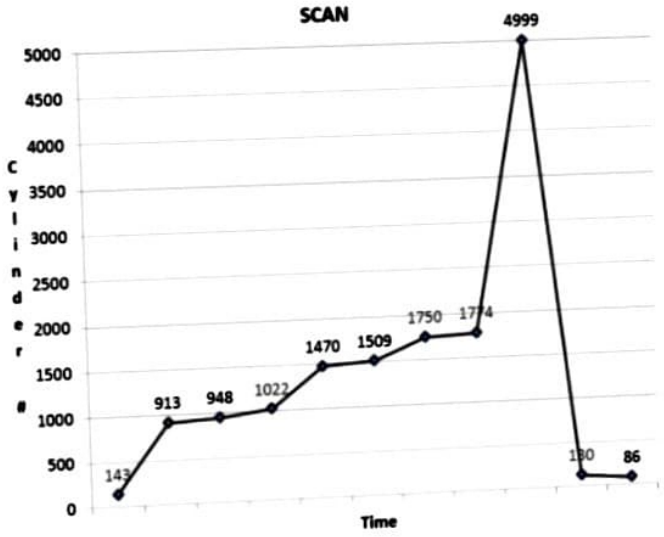
(iii) Character

9. Suppose we have files F1 to F4 in sizes of 7178, 572, 499 and 1195 bytes.Our disks have fixed physical block size of 512 bytes for allocation. Explain how many physical blocks would be needed to store these four files if we were to use a chained allocation strategy assuming that we need 5 bytes of information to determine the next block in the link. Which file results in the maximum internal fragmentation (measured as a percentage of the file size itself).

10 Consider that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving requests at cylinder 143, and the previous request was at cylinder 125. 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 pending requests for each of the following disk scheduling algorithms. A. FCFS B. SSTF C. SCAN D. C-SCAN E. LOOK F. C-LOOK

The FCFS schedule is 143, 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130.
The total seek distance is 7081.
The SSTF schedule is 143, 130, 86, 913, 948, 1022, 1470, 1509, 1750, 1774.
The total seek distance is 1745.



Total Distance travelled =
`|143-913| + |913-948| + |948-1022| + |1022-1470| + |1470-1509| + |1509-1750| + | 1750-1774 | + |1774-4999| + |4999-180| + |180-86|`
`= 9769`

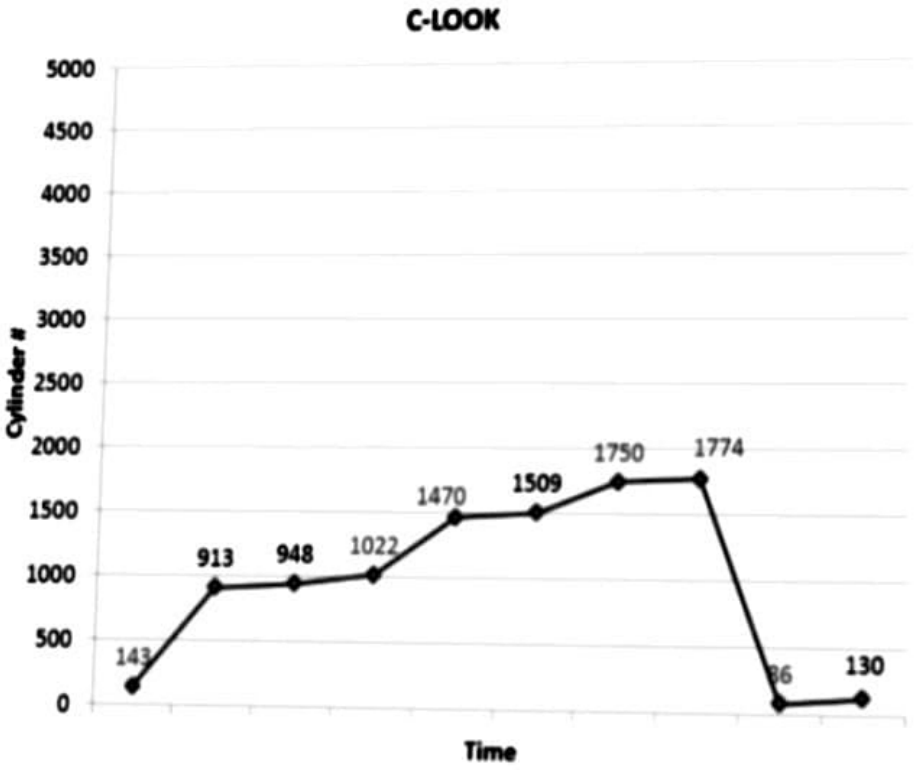
The SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774,4999, 130, 86. The total seek distance is 9769.
The LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 130, 86.
The total seek distance is 3319.

The C-SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 4999, 0, 86, 130. The total seek distance is 9985.

Total distance travelled =

|143-913| + |913-948| + |948-1022| + |1022-1470| + |1470-1509| + |1509-1750| + |1 750-1774| + |1774-86| + |86-130|

= 3363



Total distance travelled=

|143-913| + |913-948| + |948-1022| + |1022-1470| + |1470-1509| + |1509-1750| + |1 750-1774| + |1774-86| + |86-130|

= 3363

The C-LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 86, 130. The total seek distance is 3363.

PART B

1. Explain in detail the interrupts and interrupt handling features.

Interrupt: It is an event that alters the sequence in which the processor executes.

Interrupt handler: It is the function that the Kernel runs in response to a specific interrupt.

- Each device that generates interrupts has an associated interrupt handler.
- The interrupt handler for a device is a part of the device drivers.
- In Linux, interrupt handlers are normal C functions, which match a specific prototype and thus enables the Kernel to pass the handler information in standard way.
- The difference between interrupt handles from other Kernel functions is that the Kernel invokes them in response to interrupts and that they run in a special context called interrupt context.
- An interrupt can occur at any time, an interrupt handler can be executed at any time.
- It is imperative that the handler runs quickly, to resume execution of the interrupted code as soon as possible.
- It is important that

  To the hardware: The OS services the interrupt without delay.

  To the rest of system: The interrupt handler executes in as short a period as possible.

2. Discuss about Disk space management.

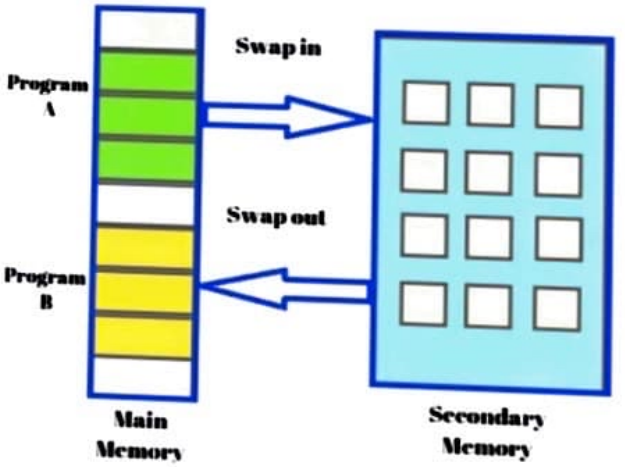
In order to ensure that your system and temporary partitions have enough storage space, you must manage your disk space. You need to maintain separate file systems, provide sufficient disk space, use shared devices for storage and manage temporary files in order to manage your disk space.

Disks serve as the main medium for storing files. A file can be defined as a collection of related information stored on a secondary device under a name. Means, a file can be viewed as a collection of bytes stored in certain memory locations. Since all these bytes represent the information related to a common specific program or job, they have to be linked together in some way or other. The simplest and easiest way to provide this linkage is to store the file in continuous memory locations.

3. Discuss about Swap – space management.

Swap-space management:

- A swap file is a space on a hard disk used as the virtual memory extension of computers real memory.
- Having a swap file allows your computer OS to pretend that you have more RAM than you actually do.
- The least recently used files in RAM can be swapped out to your hard disk until they are needed later so that new files can be swapped in to RAM.
- In larger OS the units that are moved are called pages and the swapping is called paging.



4. Describe the following Directory Implementation methods. a)Linear List b) Hash Table

Directory Implementation:

- Directories need to be fast to search, insert, and delete, with a minimum of wasted disk space.

Linear List:

- A linear list is the simplest and easiest directory structure to set up, but it does have some drawbacks.
- Finding a file (or verifying one does not already exist upon creation) requires a linear search.
- Deletions can be done by moving all entries, flagging an entry as deleted, or by moving the last entry into the newly vacant position.
- Sorting the list makes searches faster, at the expense of more complex insertions and deletions.
- A linked list makes insertions and deletions into a sorted list easier, with overhead for the links.
- More complex data structures, such as B-trees, could also be considered.

Hash Table:

- A hash table can also be used to speed up searches.
- Hash tables are generally implemented in addition to a linear or other structure.

5. Explain the concept of file sharing. What are the criteria to be followed in systems which implement file sharing.

File sharing:

It is the practice of sharing or offering access to digital information or resources, including documents, multimedia, graphics, computer programs, images and e-books. It is the private or public distribution of data or resources in a network with different levels of sharing privileges.

- File-sharing tasks use two basic sets of network criteria.

(i) Peer-to-Peer File Sharing: This is the most popular, but controversial, method of file sharing because of the use of peer-to-peer software.

- Peer-to-Peer file sharing allows users to directly access, download and edit files.
- Some 3rd party softwares facilitates Peer-to-Peer sharing by collecting large files into smaller pieces.

(ii) File Hosting Services: This is an alternative for peer-to-peer and it is a popular online material.

- These services are quite often used with Internet Collaboration Methods, including email, blogs etc., where direct download links from the file hosting services can be included.

6. Explain the following file concepts: a) File attributes b) File operations

File Attributes

Different OSes keep track of different file attributes, including:

- Name - Some systems give special significance to names, and particularly extensions (.exe, .txt, etc.), and some do not. Some extensions may be of significance to the OS (.exe), and others only to certain applications (.jpg)
- Identifier (e.g. inode number)
- Type - Text, executable, other binary, etc.
- Location - on the hard drive.
- Size
- Protection
- Time & Date
- User ID

Operations

file ADT supports many common operations:

- Creating a file
- Writing a file
- Reading a file
- Repositioning within a file
- Deleting a file
- Truncating a file.

Most OS's require that files be opened before access and closed after all access is complete. Normally the programmer must open and close files explicitly, but some rare systems open the file automatically at first access. Information about currently open files is stored in an open file table, containing for example:

- File pointer - records the current position in the file, for the next read or write access.
- File-open count - How many times has the current file been opened (simultaneously by different processes) and not yet closed? When this counter reaches zero the file can be removed from the table. o Disk location of the file.
- Access rights: Some systems provide support for file locking.
- A shared lock is for reading only.
- An exclusive lock is for writing as well as reading.
- An advisory lock is informational only, and not enforced. (A "Keep Out" sign, which may be ignored.)
- A mandatory lock is enforced. (A truly locked door.) o UNIX used advisory locks, and Windows used mandatory locks.

7. Explain the following file concepts: a) File types b) Internal file structure

File Types:

There are a large number of file types. Each has a particular purpose. The type of a file indicates its use cases, contents, etc. Some common types are:

1. Media:
   Media files store media data such as images, audio, icons, video, etc. Common extensions: img, mp3, mp4, jpg, png, flac, etc.

2. Programs:
   These files store code, markup, commands, scripts, and are usually executable. Common extensions: c, cpp, java, xml, html, css, js, ts, py, sql, etc.

3. Operating System Level:
   These files are present with the OS for its internal use. Common extensions: bin, sh, bat, dl, etc.

4. Document:
   These files are used for managing office programs such as documents, spreadsheets, etc. Common extensions: xl, doc, docx, pdf, ppt, etc.

5. Miscellaneous:
   Generic text file(.txt), canvas files, proprietary files, etc.

Internal File Structure

- Disk files are accessed in units of physical blocks, typically 512 bytes or some power-of-two multiple thereof. (Larger physical disks use larger block sizes, to keep the range of block numbers within the range of a 32-bit integer.)

- Internally files are organised in logical units, which may be as small as a single byte, or may be a larger size corresponding to some data record or structure size.
- The number of logical units which fit into one physical block determines its packing, and has an impact on the amount of internal fragmentation (wasted space) that occurs.
- As a general rule, half a physical block is wasted for each file, and the larger the block sizes the more space is lost to internal fragmentation.

8. Discuss the following a) File system mounting b)Thrashing

a) Before you can access the files on a file system, you need to mount the file system. Mounting a file system attaches that file system to a directory (mount point) and makes it available to the system. The root (/) file system is always mounted. Any other file system can be connected or disconnected from the root (/) file system.
When you mount a file system, any files or directories in the underlying mount point directory are unavailable as long as the file system is mounted. These files are not permanently affected by the mounting process, and they become available again when the file system is unmounted. However, mount directories are typically empty, because you usually do not want to obscure existing files.

b) Thrashing is computer activity that makes little or no progress, usually because memory or other resources have become exhausted or too limited to perform needed operations. When this happens, a pattern typically develops in which a request is made of the operating system by a process or program, the operating system tries to find resources by taking them from some other process, which in turn makes new requests that can't be satisfied. In a virtual storage system (an operating system that manages its

logical storage or memory in units called pages), thrashing is a condition in which excessive paging operations are taking place.
A system that is thrashing can be perceived as either a very slow system or one that has come to a halt.

9. Explain caching.

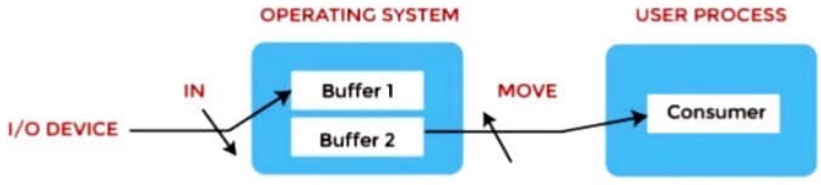
- Caching transparently stores data in a component called Cache, so that future requests for that data can be served faster.
- A special high-speed storage mechanism. It can be either a reserved section of main memory or an independent high-speed storage device.
- The data that is stored within a cache might be values that have been computed earlier or duplicates of original values that are stored elsewhere.
- E.g: Memory Caching, Disk Caching, Web Caching(used in browser), Database Caching etc.
- The data which is to be used many times results in wastage of time if it is in hard disk, but storing the data in cache reduces this time wastage.

A disk cache is a mechanism for improving the time it takes to read from or write to a hard disk. Today, the disk cache is usually included as part of the hard disk. A disk cache can also be a specified portion of random access memory (RAM). The disk cache holds data that has recently been read and, in some cases, adjacent data areas that are likely to be accessed next. Write caching is also provided with some disk caches.

10. Define buffering.

- Preloading data into a reserved area of memory (the buffer).

- It temporarily stores input or output data in an attempt to better match the speeds of two devices such as a fast CPU and a slow disk drive.
- Buffers may be used in between when moving data between two processes within a computer. Data is stored in a buffer as it is retrieved from one process or just before it is sent to another process.
- With spooling, the disk is used as a very large buffer. Usually complete jobs are queued on disk to be completed later.
- It is mostly used for input, output, and sometimes temporary storage of data either when transfer of data takes place or data that may be modified in a non-sequential manner.

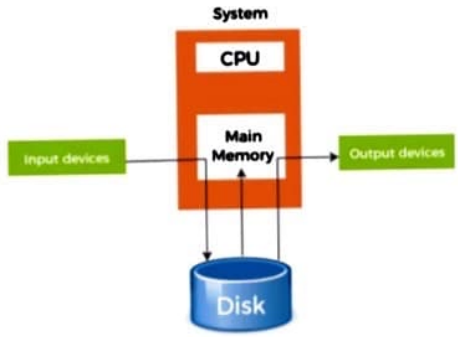


DOUBLE I/O BUFFERING

11. Write about spooling.

- Acronym for "Simultaneous Peripheral Operation On-Line".
- It's a process of placing data in a temporary working area for another program to process. E.g: Print spooling and Mail spools etc.
- When there is a resource (like printer) to be accessed by two or more processes(or devices), spooling comes handy to schedule the tasks. Data from each process is put on the spool (print queue) and processed in FIFO(first in first out) manner.
- With spooling all processes can access the resource without waiting.
- After writing the data on a spool, the process can perform other tasks. And the printing process operates separately.
- Without spooling, the process would be tied up until the printing finished.

- Spooling is useful for devices which have differing data access rates. Used mainly when processes share some resource and need to have synchronisation.



12. Explain the techniques used for performing I/O Operations.

Programmed I/O: The processor issues an I/O command, on behalf of a process, to an I/O module; that process then busy-waits for the operation to be completed before proceeding.

Interrupt-driven I/O: The processor issues an I/O command on behalf of a process, continues to execute subsequent instructions, and is interrupted by the I/O module when the latter has completed its work. The subsequent instructions may be in the same process, if it is not necessary for that process to wait for the completion of the I/O. Otherwise, the process is suspended pending the interrupt and other work is performed.

Direct memory access (DMA): A DMA module controls the exchange of data between main memory and an I/O module. The processor sends a request for the transfer of a block of data to the DMA module and is interrupted only after the entire block has been transferred.

13. Give an example of an application in which data in a file should be accessed in the following order: i. Sequential ii. Random.

Sequential Access to a data file means that the computer system reads or writes information to the file sequentially, starting from the beginning of the file and proceeding step by step.

On the other hand, Random Access to a file means that the computer system can read or write information anywhere in the data file.

A more modern example is a **cassette tape** (sequential—you have to fast-forward through earlier songs to get to later ones) and a CD (random access—you can skip to the track you want).

14. Explain the following in detail with respect to the disk. a) Seek time b) Latency.

Seek time:

- A disk is divided into many circular tracks.
- Seek time is defined as the time required by the read/write head to move from one track to another.

Latency:

- The disk is divided into many circular tracks and these tracks are further divided into blocks known as sectors.
- The time required by the read/write head to rotate to the requested sector from the current position is called Rotational Latency.

15. Explain the following in detail with respect to disk. a)Access time b) Transfer time.

Access time : It is defined as the setup time before the actual data transfer takes place.

Access time is the summation of

a) Seek time

b) Rotational latency

c) Command processing time

d) Settle time

NOTE: Command processing time and settle time are taken as zero.

Transfer time: It is defined as the time required to transfer data between the system and the disk.

Transfer time is of two types:

a) Internal transfer rate

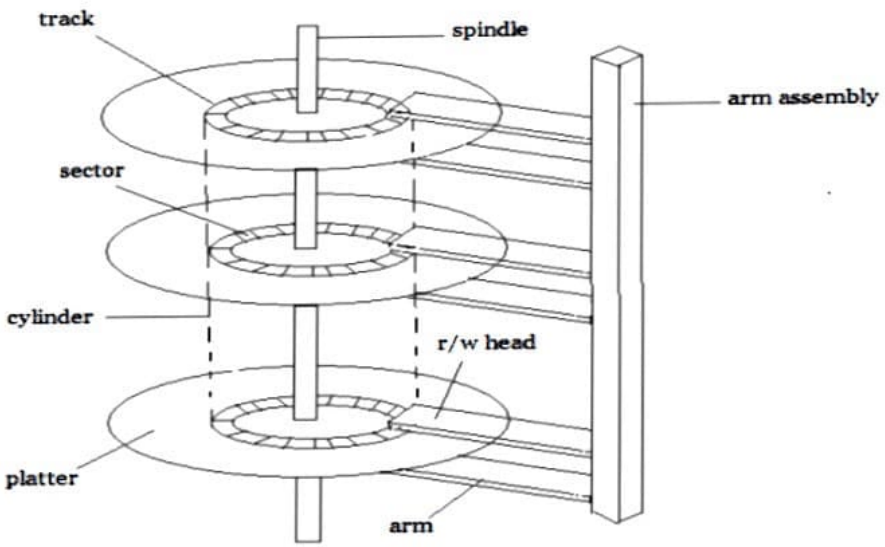
b) External transfer rate

a) Internal transfer rate: It is defined as the time required to move data between disk surface and hard disk cache.

b) External transfer rate: It is defined as the time required to move data between hard disk cache and system.

16. Define magnetic disk structure and its management.

In modern computers, most of the secondary storage is in the form of magnetic disks. Hence, knowing the structure of a magnetic disk is necessary to understand how the data in the disk is accessed by the computer.



Structure of a magnetic disk

A magnetic disk contains several platters. Each platter is divided into circular shaped tracks. The length of the tracks near the center is less than the length of the tracks farther from the center. Each track is further divided into sectors, as shown in the figure.

Tracks of the same distance from the center form a cylinder. A read-write head is used to read data from a sector of the magnetic disk.

The speed of the disk is measured as two parts:

- Transfer rate: This is the rate at which the data moves from disk to the computer.
- Random access time: It is the sum of the seek time and rotational latency.

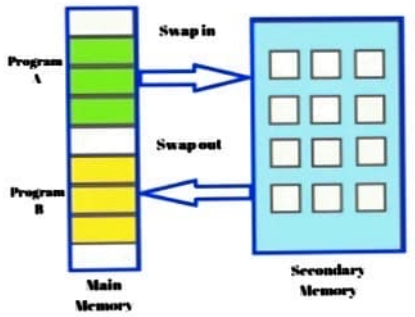
  Seek time is the time taken by the arm to move to the required track.

  Rotational latency is defined as the time taken by the arm to reach the required sector in the track.

  Even though the disk is arranged as sectors and tracks physically, the data is logically arranged and addressed as an array of blocks of fixed size. The size of a block can be 512 or 1024 bytes. Each logical block is mapped with a sector on the disk, sequentially. In this way, each sector in the disk will have a logical address.

17. Explain swap space management.

A swap file (or swap space or, in Windows NT, a pagefile) is a space on a hard disk used as the virtual memory extension of a computer's real memory (RAM). Having a swap file allows your computer's operating system to pretend that you have more RAM than you actually do. The least recently used files in RAM can be "swapped out" to your hard disk until they are needed later so that new files can be "swapped in" to RAM. In larger operating systems (such as IBM's OS/390), the units that are moved are called pages and the swapping is called paging.



18. Differentiate among the following disk scheduling algorithms. a) FCFS b) SSTF

1. FCFS: FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.

Advantages:

- Every request gets a fair chance
- No indefinite postponement

Disadvantages:

- Does not try to optimise seek time
- May not provide the best possible service

2. SSTF: In SSTF (Shortest Seek Time First), requests having the shortest seek time are executed first. So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first. SSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of the system.

Advantages:

- Average Response Time decreases
- Throughput increases

Disadvantages:

- Overhead to calculate seek time in advance.

- Can cause Starvation for a request if it has higher seek time as compared to incoming requests.
- High variance of response time as SSTF favours only some requests

Also refer Difference between FCFS and SSTF Disk Scheduling Algorithm - GeeksforGeeks

19. Differentiate among the following disk scheduling algorithms. a) SCAN b) C-SCAN

SCAN: In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works like an elevator and hence is also known as elevator algorithm. As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.

Advantages:

- High throughput
- Low variance of response time
- Average response time

Disadvantages:

- Long waiting time for requests for locations just visited by disk arm

CSCAN: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.

These situations are avoided in the SAN algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to the SCAN algorithm and hence it is known as C-SCAN (Circular SCAN).

Advantages:

- Provides more uniform wait time compared to SCAN

Also refer Difference between SCAN and CSCAN Disk scheduling algorithms - GeeksforGeeks

20. Differentiate among the following disk scheduling algorithms. a)LOOK b) C-LOOK

LOOK: It is similar to the SCAN disk scheduling algorithm except the difference is 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.

CLOOK: As LOOK is similar to SCAN algorithm, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, 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.