Solved Questions Module-5

1. Compare sequential access and direct access methods of storage devices.

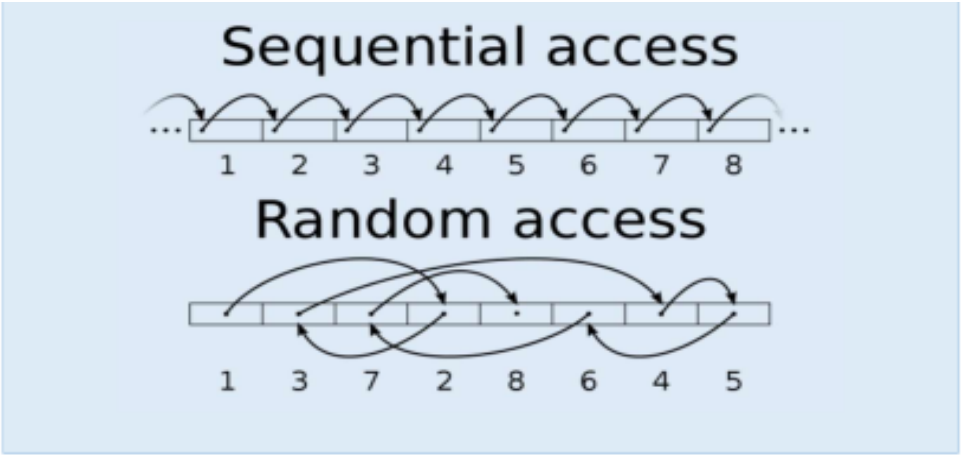
FILE ACCESS METHODS

❖**Sequential access**

- The records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other. This access method is the most primitive one. Example: Compilers usually access files in this fashion.
- In sequential access, the OS read the file word by word. A pointer is maintained which initially points to the base address of the file. If the user wants to read first word of the file then the pointer provides that word to the user and increases its value by 1 word. This process continues till the end of the file

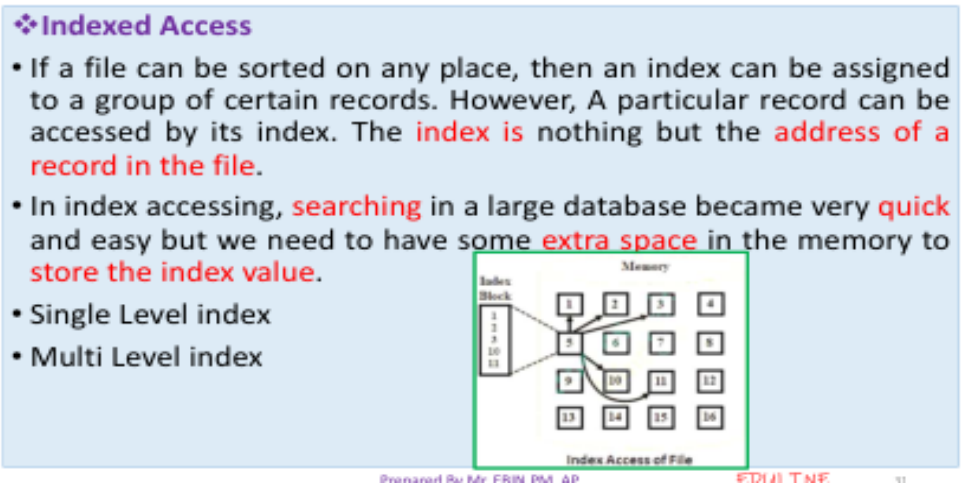
❖**Direct / Random Access**

- The Direct Access is mostly required in the case of database systems. In most of the cases, we need filtered information from the database. The sequential access can be very slow and inefficient in such cases.
- Suppose every block of the storage stores 4 records and we know that the record we needed is stored in 10th block. In that case, the sequential access will not be implemented because it will traverse all the blocks in order to access the needed record.
- Direct access will give the required result despite of the fact that the operating system has to perform some complex tasks such as determining the desired block number. However, that is generally implemented in database applications.



Mr. EBIN PM, AP, Chandigarh

SYSTEMS http://www.youtube.com/c/EDULI



2. Write notes on disk formatting.

3. How is disk formatting done?

Disk Formatting

- A new magnetic disk is a blank slate: it is just a platter of a magnetic recording material. Before a disk can store data, it must be divided into sectors that the disk controller can read and write. This process is called low-level formatting,or physical formatting.
- Low-level formatting fills the disk with a special data structure for each sector. The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.
- The header and trailer contain information used by the disk controller, such as a sector number and an error-correcting code (ECC).
- When the controller writes a sector of data during normal I/O, the ECC is updated with a value calculated from all the bytes in the data area. When the sector is read, the ECC is recalculated and compared with the stored value. If the stored and calculated numbers are different, the data area of the sector has become corrupted and that the disk sector may be bad.
- The ECC is an error-correcting code. If only a few bits of data have been corrupted, ECC enable the controller to identify which bits have been changed and calculate what their correct values should be.
- It then reports a recoverable soft error. The controller automatically does the ECC processing whenever a sector is read or written.
- For many hard disks, when the disk controller is instructed to low-level-format the disk, it can also be told how many bytes of data space to leave between the header and trailer of all sectors.
- It is usually possible to choose among a few sizes, such as 256, 512, and 1,024 bytes. Formatting a disk with a larger sector size means that fewer sectors can fit on each track; but it also means that fewer headers and trailers are written on each track and more space is available for user data.
- Some operating systems can handle only a sector size of 512 bytes.
- The first step is to partition the disk into one or more groups of cylinders. The operating system can treat each partition as though it were a separate disk.
- The second step is logical formatting, or creation of a file system. In this step, the operating system stores the initial file-system data structures onto the disk.
- To increase efficiency, most file systems group blocks together into larger chunks, frequently called clusters.
- Some operating systems give special programs the ability to use a disk partition as a large sequential array of logical blocks, without any file-system data structures.

4. Define seek time, rotational latency and disk bandwidth of disks.

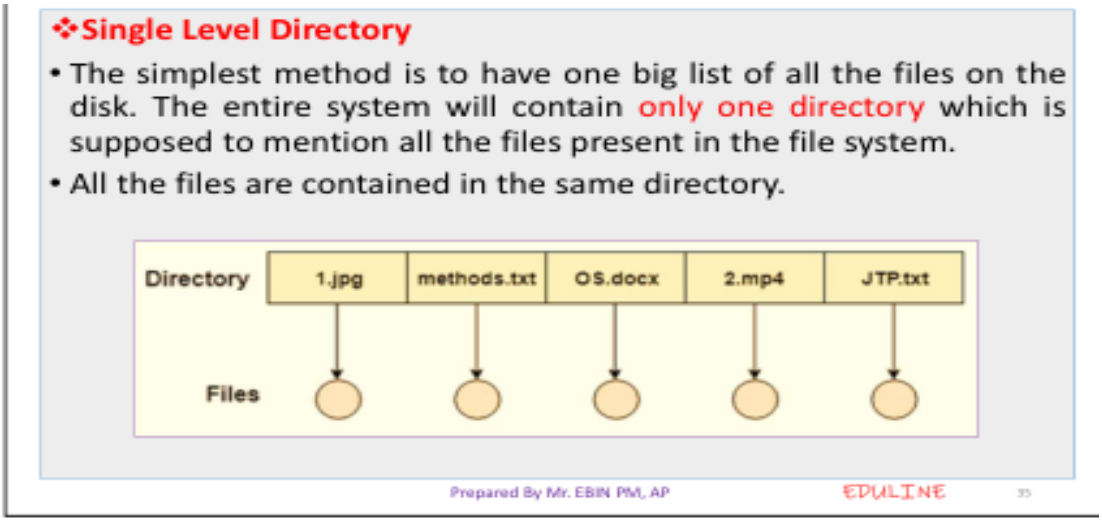
5. Define the terms seek time, rotational latency and transfer time

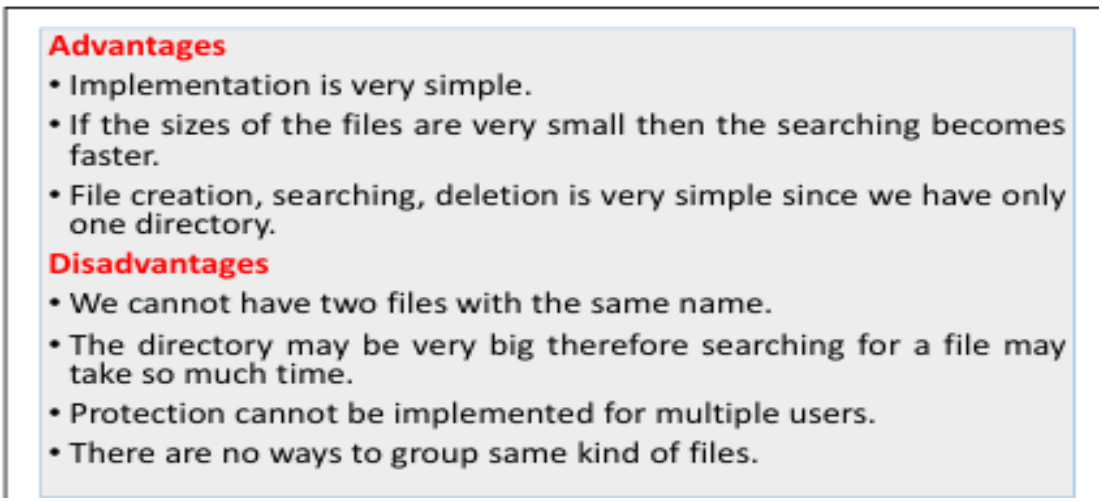
6. Define the terms i) Seek time ii) Rotational delay iii) Disk bandwidth.

- The seek time is the time for the disk arm to move the heads to the cylinder containing the desired sector.
- The rotational latency is the additional time for the disk to rotate the desired sector to the disk head.
- 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.
- Transfer time is the time taken to transfer the data.

7. Explain two level directory structure with the help of a diagram.

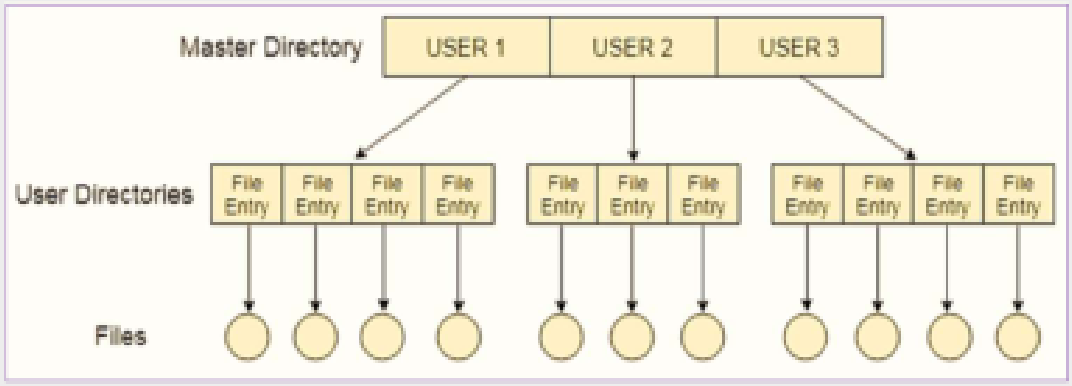
8. Explain single level directory structure with an example.

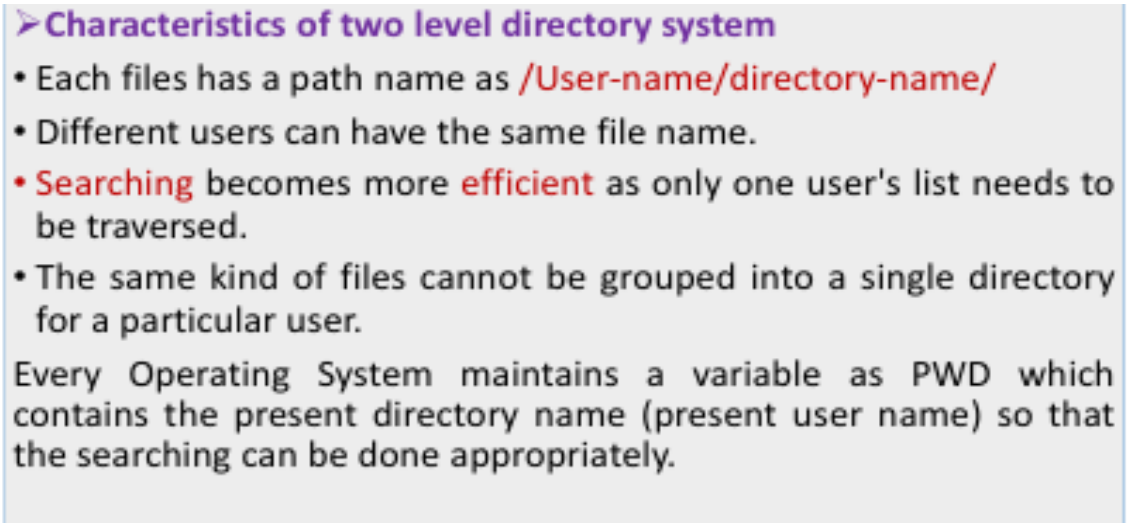




❖Two Level Directory

- In two level directory systems, we can create a separate directory for each user.
- There is one master directory which contains separate directories dedicated to each user.
- For each user, there is a different directory present at the second level, containing group of user's file.
- The system doesn't let a user to enter in the other user's directory without permission.





Single directory structure

• The single-level directory is the simplest directory structure.

• All files are contained in the same directory

• Since all the files are in the same directory, they must have a unique name.

• Searching will become time taking if the directory is large.

• This can't group the same type of files together**.**

9. Briefly explain about file attributes.

A file's attributes typically consist of these:

 Name. The symbolic file name is the only information kept in human- readable form.

 Identifier. This unique tag, usually a number, identifies the file within the file system; it is the non-human-readable name for the file.

 Type. This information is needed for systems that support different types of files.

 Location. This information is a pointer to a device and to the location of the file on that device.

 Size. The current size of the file (in bytes, words, or blocks) and possibly the maximum allowed size are included in this attribute.

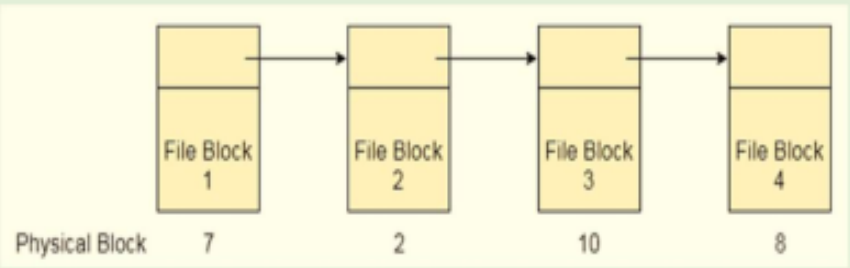
 Protection. Access-control information determines who can do reading, writing, executing, and so on.

 Time, date, and user identification. This information may be kept for creation, last modification, and last use. These data can be useful for protection, security, and usage monitoring.

10. Explain the linked allocation of file allocation with merits and demerits.

2. Linked List Allocation

- Linked List allocation solves all problems of contiguous allocation. In linked list allocation, each file is considered as the linked list of disk blocks. However, the disks blocks allocated to a particular file need not to be contiguous on the disk. Each disk block allocated to a file contains a pointer which points to the next disk block allocated to the same file.



Advantages

- There is no external fragmentation with linked allocation.
- Any free block can be utilized in order to satisfy the file block requests.
- File can continue to grow as long as the free blocks are available.
- Directory entry will only contain the starting block address.

Disadvantages

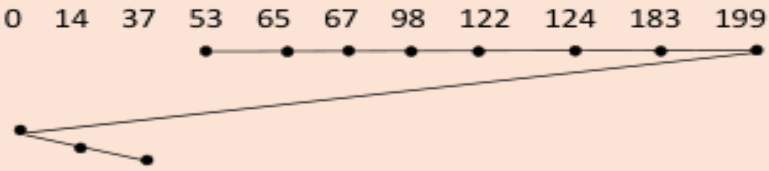
- Random Access is not provided.
- Pointers require some space in the disk blocks.
- Any of the pointers in the linked list must not be broken otherwise the file will get corrupted.
- Need to traverse each block.

11. Point out the differences between C-SCAN and C-LOOK scheduling algorithms.

➢C – SCAN SCHEDULING

- In C-SCAN algorithm, the arm of the disk moves in a particular direction servicing requests until it reaches the last cylinder, then it jumps to the last cylinder of the opposite direction without servicing any request then it turns back and start moving in that direction servicing the remaining requests.

- Eg: Queue = 98,183,37,122,14,124,65,67 . Head starts at 53.

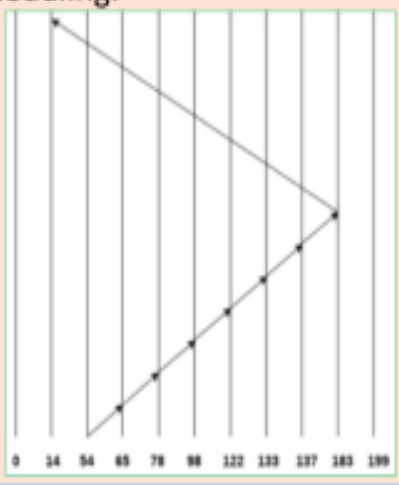


No. of head movements = (199-53)+37 = **183**

➢C Look Scheduling

- C Look Algorithm is similar to C-SCAN algorithm to some extent.
- In this algorithm, the arm of the disk moves outwards servicing requests until it reaches the highest request cylinder, then it jumps to the lowest request cylinder without servicing any request then it again start moving outwards servicing the remaining requests.
- It is different from C SCAN algorithm in the sense that, C SCAN force the disk arm to move till the last cylinder regardless of knowing whether any request is to be serviced on that cylinder or not.

Eg: Consider the following disk request sequence for a disk with 100 tracks 98, 137, 122, 183, 14, 133, 65, 78 . Head pointer starting at 54 and moving in left direction. Find the number of head movements in cylinders using C LOOK scheduling.

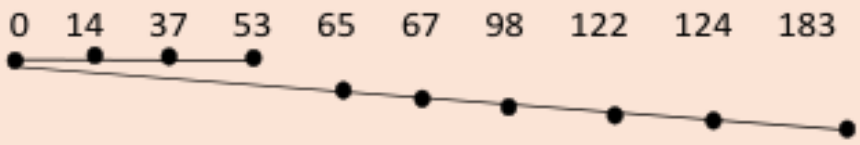


Number of head movements = 11 + 13 + 20 + 24 + 11 + 4 + 46 + 169 = **298**

12. Explain “Elevator” algorithm for disk scheduling with example.

➢SCAN SCHEDULING

- It is also called as Elevator Algorithm.
- In this algorithm, the disk arm moves into a particular direction till the end, satisfying all the requests coming in its path, and then it turns back and moves in the reverse direction satisfying requests coming in its path.
- It works in the way an elevator works, elevator moves in a direction completely till the last floor of that direction and then turns back.
- Eg: queue = 98,183,37,122,14,124,65,67. Head starts at 53. Assume that the disk arm is moving towards left(to zero)



No. of head movements = 53+183 = **236**

13. Write short note on protection domain.

6.3 Domain of Protection

- A computer system is a collection of processes and objects. By objects, we mean both hardware objects (such as the CPU, memory segments, printers, disks, and tape drives) and software objects (such as files, programs, and semaphores). Each object has a unique name that differentiates it from all other objects in the system, and each can be accessed only through well-defined and meaningful operations. Objects are essentially abstract data types.
- The operations that are possible may depend on the object.
- A process should be allowed to access only those resources for which it has authorization.
- At any time, a process should be able to access only those resources that it currently requires to complete its task. This second requirement, commonly referred to as the need-to-know principle, is useful in limiting the amount of damage a faulty process can cause in the system.

1. Domain Structure

- A protection domain, which specifies the resources that the process may access. Each domain defines a set of objects and the types of operations that may be invoked on each object.
- The ability to execute an operation on an object is an accessright. A domain is a collection of access rights, each of which is an ordered pair <object-name, rights- set>.
- For example, if domain D has the access right <file F, {read,write}>, then a process executing in domain D can both read and write file F. It cannot, however, perform any other operation on that object.
- Domains may share access rights. Ex. we have three domains: D1, D2,and D3. The access right <O4, {print}> is shared by D2 and D3, implying that a process executing in either of these two domains can print object O4.
- The association between a process and a domain may be either static, if the set of resources available to the process is fixed throughout the process's lifetime, or dynamic.
- If the association between processes and domains is fixed, and we want to adhere to the need-to-know principle, then a mechanism must be available to change the content of a domain.

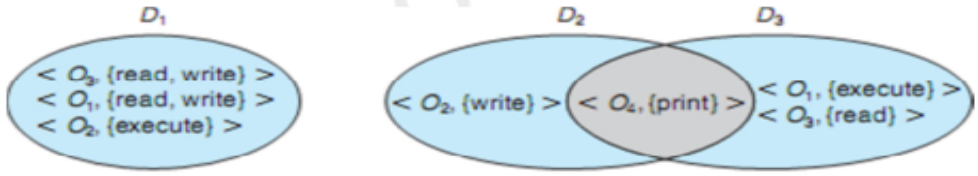


Figure 14.1 System with three protection domains.

- If the association is dynamic, a mechanism is available to allow domain switching, enabling the process to switch from one domain to another. We may also want to allow the content of a domain to be changed. If we cannot change the content of a domain, we can provide the same effect by creating a new domain with the changed content and switching to that new domain when we want to change the domain content.
- A domain can be realized in a variety of ways:
    1. Each user may be a domain.
    2. Each process may be a domain.
    3. Each procedure may be a domain.