PES UNIVERSITY
(Established under Karnataka Act No.16 of 2013)
100-ft Ring Road, BSK III Stage, Bangalore – 560 085
Department of Computer Science & Engg
Session: Jan-May 2021
UE19CS254: Operating Systems
UNIT 4 Question Bank: (Solutions to selected questions)

#	
	Chapter 10: File System
1.	Consider a file system where a file can be deleted and its disk space reclaimed while links to that file still exist. What problems may occur if a new file is created in the same storage area or with the same absolute path name? How can these problems be avoided? **Answer:** When the new file takes the place of the old file, the links to the old file will now point to the new file. Users might think they are accessing the old file when in fact they are accessing the new file. Depending on the file system, the protection mode for the old file might get used instead of the protection mode for the new file. The solution is to remove all links to a file when it is deleted. One way to do this is to keep a list of links to a file with the file data. Then when the file is deleted, the links can easily be deleted as well. Another way to do this is to keep a reference count (as does the Unix inode) and only delete the file data when the reference count goes to 0.
2.	Explain in detail the Various File Access Methods. **Answer:** The two methods of accessing the information in a file are Sequential and Direct. Sequential access is the method by which systems process information in order, one record after the other. A read operation read_next() reads the next portion of the file and automatically advances the file pointer. Similarly, write_next() appends to the end of the file and advances pointer to the new end of the file. Sequential access is based on the tape model of a file and works as well on sequential-access devices as it does on random-access ones. The other method is direct access (or relative access). A file is considered to be made up of fixed-length logical records that allow programs to read and write records in no particular order. The direct-access method is based on the disk model of a file, since disks allow random access to any file block. Here, the file operations must be modified to include the block number as a parameter. Thus, we have read(n) and write(n). The block number provided by the user to the operating system is generally a relative block number.
3.	Explain the various logical structures of directories. List the advantages and disadvantages of each structure. **Answer:**

Single-level directory
The simplest directory structure. All files are contained in the same directory which makes it easy to implement and understand.
Advantages:

- Since it is a single directory, its implementation is simple.
- If the files are smaller in size, searching will be faster.
- Operations like file creation, searching, deletion, updating can be easily performed in such a directory structure.

Disadvantages:

- No two files can have the same name, it will lead to a name collision.
- If the directory is large, then searching will be time consuming.
- No grouping capability.

Two-level directory
Each user has their own *user files directory (UFD)*. The UFDs have similar structures, but each lists only the files of a single user.
Advantages:

- Different users can have the same directory and file names.
- Searching of files becomes easier using pathname and user-grouping.

Disadvantages:

- No sharing of files amongst users.
- No grouping capability inside each user.

Tree-structured directory
A two-level directory is a tree of height 2, the natural generalization is to extend the directory structure to a tree of arbitrary height.
This generalization allows the user to create their own subdirectories and to organize their files accordingly. The tree has a root directory, and every file in the system has a unique path.
Advantages:

- Scalable as the probability of name collision is less.
- Searching is easy as both absolute and relative path can be used.

Disadvantages:

- Every file may not fit into the hierarchical model, files may need to be saved into multiple directories.
- No sharing of files.

Acyclic Graph Directory
An acyclic graph is a graph with no cycle and this allows us to share subdirectories and files. The same file or subdirectories may be in two different directories. If any user makes some changes in the shared file, it will reflect in all parent subdirectories.
Advantages:

- Sharing of files.
- No restrictions on the structure of files and directories.

Disadvantages:

- Sharing of files is implemented by linking.
- If the link is a soft link then after deleting the shared file, we are left with a dangling pointer.

	• In the case of a hard link, to delete a shared file we have to delete all the references associated with it. General Graph Directory Here, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory. The main problem with this kind of directory structure is to calculate the total size or space that has been taken by the files and directories. Advantages: • It allows cycles. • It is more flexible than other directories structure. Disadvantages: • It is more costly than others. • It needs garbage collection. (For Diagrams of all directory structures, refer section 10.3 of text book.)
Chapter 11: Implementing File Systems	
1.	What are the different methods of allocation in a File System? **Answer:** 1. Contiguous Allocation Here, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be: *b, b+1, b+2,……b+n-1.* This means that given the starting block address and the length of the file (in terms of blocks required), we can determine the blocks occupied by the file. The directory entry for a file with contiguous allocation contains • Address of starting block • Length of the allocated portion. Advantages: • Both the Sequential and Direct Access are supported by this. • Fast since the number of seeks are minimal because of contiguous allocation of file blocks. Disadvantages: • Suffers from both internal and external fragmentation. This makes it inefficient in terms of memory utilization. • Increasing file size is difficult because it depends on the availability of contiguous memory at a particular instance. 2. Linked List Allocation In this scheme, each file is a linked list of disk blocks which need not be contiguous. The disk blocks can be scattered anywhere on the disk. The directory entry contains a pointer to the starting and the ending file block. Each block contains a pointer to the next block occupied by the file. Advantages: • Flexible in terms of file size as system does not have to look for a contiguous chunk of memory. • Does not suffer from external fragmentation. Disadvantages:

	• Because the file blocks are distributed randomly on the disk, a large number of seeks are needed to access every block individually. This makes linked allocation slower. • It does not support random or direct access. • Pointers required incur some extra overhead. 3. Indexed Allocation In this scheme, a special block known as the Index block contains the pointers to all the blocks occupied by a file. Each file has its own index block. The ith entry in the index block contains the disk address of the ith file block. Advantages: • Supports direct access to blocks occupied by the file and therefore provides fast access to the file blocks. • No external fragmentation. Disadvantages: • The pointer overhead for indexed allocation is greater than linked allocation. • For very small files, say files that extend to only 2-3 blocks, indexed allocation would keep one entire block (index block) for the pointers which is inefficient in terms of memory utilization. However, in linked allocation we lose the space of only 1 pointer per block. (For diagrams of the allocation schemes, refer section 11.4 of text book.)
Chapter 12: Mass-Storage Structure	
1.	None of the disk-scheduling disciplines, except FCFS, is truly fair (starvation may occur). a. Explain why this assertion is true. b. Describe a way to modify algorithms such as SCAN to ensure fairness. c. Explain why fairness is an important goal in a time-sharing system. d. Give three or more examples of circumstances in which it is important that the operating system be unfair in serving I/O requests. **Answer:** (a) New requests for the track over which the head currently resides can theoretically arrive as quickly as these requests are being serviced. (b) All requests older than some predetermined age could be forced to the top of the queue, and an associated bit for each could be set to indicate that no new request could be moved ahead of these requests. For SSTF, the rest of the queue would have to be reorganized with respect to the last of these old requests. (c) To prevent unusually long response times. (d) Paging and swapping should take priority over user requests. It may be desirable for other kernel initiated I/O, such as the writing of file system metadata, to take precedence over user I/O. If the kernel supports real-time process priorities, the I/O requests of those processes should be favoured.
2.	Compare the throughput achieved by a RAID level 5 organization with that achieved by a RAID level 1 organization for the following: a. Read operations on single blocks b. Read operations on multiple contiguous blocks

	Answer: a) The amount of throughput depends on the number of disks in the RAID system. A RAID Level 5 comprising of a parity block for every set of four blocks spread over five disks can support four to five operations simultaneously. A RAID Level 1 comprising of two disks can support two simultaneous operations. Of course, there is greater flexibility in RAID Level 1 as to which copy of a block could be accessed and that could provide performance benefits by taking into account position of disk head. b) RAID Level 5 organization achieves greater bandwidth for accesses to multiple contiguous blocks since the adjacent blocks could be simultaneously accessed. Such bandwidth improvements are not possible in RAID Level 1.