Internal Assessment Test 1 – May 2022

QP Set 2

Sub:	File Structures				Sub Code:	18IS61	Branch:	ISE
Date:	06/05/2022	Duration:	90 min's	Max Marks: 50	Sem/Sec:	VI A, B & C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1.	*State the function of CD-ROM in terms of Read and Write operations.*	10	CO1	L1

Physical Organization of CD-ROM

- Reading Pits and Lands

CD-ROMs made up of class,

has a coating that is changed by the laser beam.

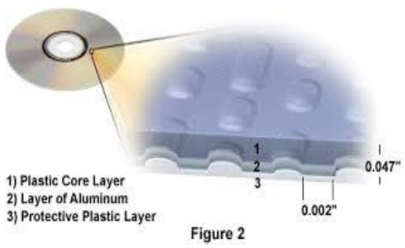
- the area hit by the laser beam turn into pits along the tracks.

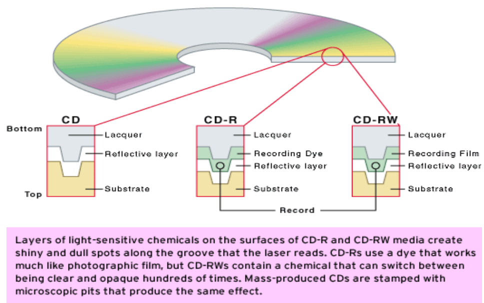
-the smooth, unchanged areas between the pits are called lands

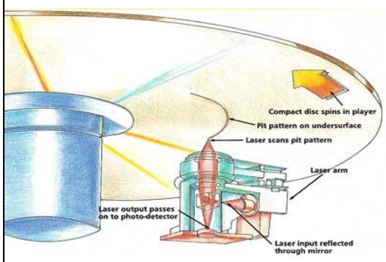
CD-ROM – Working Model

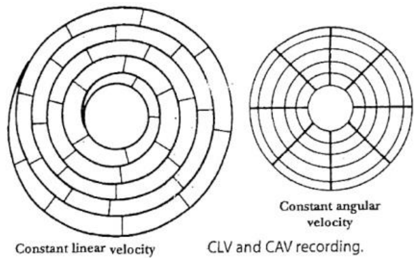
Pits – a tiny hole made on the surface of the CD. This pit will scatter the laser light

Land – the surface of the CD disk, which can reflect light back.









CLV and CAV recording.

Constant Linear Velocity - The disk has to spin more slowly when reading outer track.

Constant reading

***Constant angular Velocity -** Write data less densely in the outer tracks*

Spins the disk at same speed, Wasting storage capacity

Constant angular Velocity, Write data less densely in the outer tracks

2.	Discuss the concept of Buffer Management in details. Justify how the double buffering increase performance?	10	CO1	L2

Buffering involves working with large chunks of data in memory so the number of disk access can be reduced.

Buffer Bottlenecks

File Manager allocates I/O buffers to hold incoming data

How many buffers? File Manager allocates several buffers for performing I/O operations

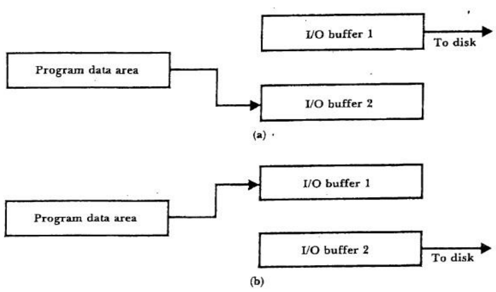
Eg., program asks (need) for its first character, and I/O buffer is loaded with the sector containing the character, and the character is transmitted to the program.

- The program output the character, the I/O buffer is filled with the sector into output buffer character need to go, destroying original contents.

Buffering strategies

Multiple Buffering – Double buffering

If two buffers are used, and CPU overlapping is permitted, CPU can be filling one buffer while the content of the other are being transmitted to disk.



'n' number of buffers can be used, and they can be organized in different ways

Some file system using buffer pooling: When a system buffer is needed, it is taken from a pool of available buffers and used it.

When a system receives a request to read a certain sector or block, it looks to see if one its buffers already contains that sector or block.

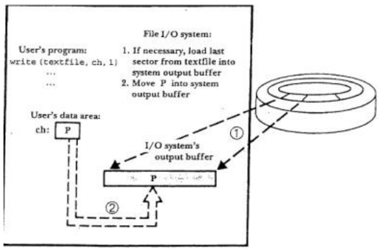
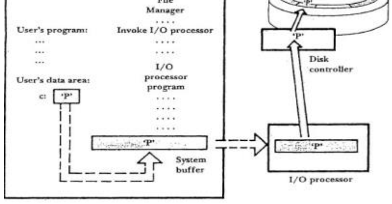
If no, system will find one buffer from pool which is currently not used .

Many algorithms are there to make use of buffer pool - FCFS, LRU

Move mode and locate mode

Scatter/Gather I/O

File with many block, and each block consists of header followed by data.

	To ease of access, all headers in each block kept in one buffer, and data in another buffer. (Whole data in single buffer)			
3.	Explain the journey of bytes with appropriate steps between program to disk and vice versa. The user wants to write a character ‘P’ into the file, and read from the file. The journey of bytes describe how the data moving from file to program and from program to file. suppose write(textfile, ch, 1) User’s program Write(textfile, ch, 1) Operating System’s file I/O system Get one byte from variable ‘ch’ in user program’s data area. Write it to current location in text file: The program asks the operating system to write the contents of the variable ‘c’ to the next available position in TEXT. The Operating System passes the job on to the file manager The file manager looks up TEXT in a table containing information about it The file manager searches a file allocation table for the physical location of the sector that is to contain the bytes The file manager makes sure that the last sector in the file has been stored in a system I/O buffer in RAM, then deposit the ‘p’ into its proper position in the buffer The file manager gives instruction to the I/O processor about where the byte is stored in RAM and where it needed to be sent on the disk The I/O processor finds a time when the drive is available to receive the data and puts the data in proper format for the disk. The I/O processor sends the data to the disk controller The controller instructs the drive to move the read/write head to the proper track then write The I/O Buffer - ‘P’ wants to write/read - The FM determines the sector that is to contain the ‘p’ is already in memory or not. - If need to load the sector, the FM must find an available system I/O buffer space for it and then read it from the disk. The Bytes Leaves Memory: The I/O Processor and Disk Controller To save CPU time, I/O processor can help   File manager instructs the I/O processor to write data on disk. The I/O Processor gets the data from the system buffer, prepare it to storing into disk.	10	CO1	L2
4.	Illustrate sequential and direct searching of records in file and justify how the direct accessing increases the file system performance. Explain with neat diagram.	10	CO1	L3

<table>
<tr><td></td><td>The sequential search takes O(n) time complexity to search the elements in either file/table.

If the location of the desired record is known, we can directly access it. Here, for direct searching, we can use relative record number (RRN). The RRN number specified where the actual record stored on the disk. This RRN number is stored along with key which is used search.</td><td></td><td></td><td></td></tr>
<tr><td>5.</td><td>Discuss about to handle free space of variable-length and fixed-length records.

- space reclamation – fixed-length Record

That deleted records are marked in some special way, and

We can find the space that deleted records once occupied

Sequential search – through a file – is a slow process

A way to know immediately if there are empty slots in the file

A way to jump directly to one those slots …

Linked Lists

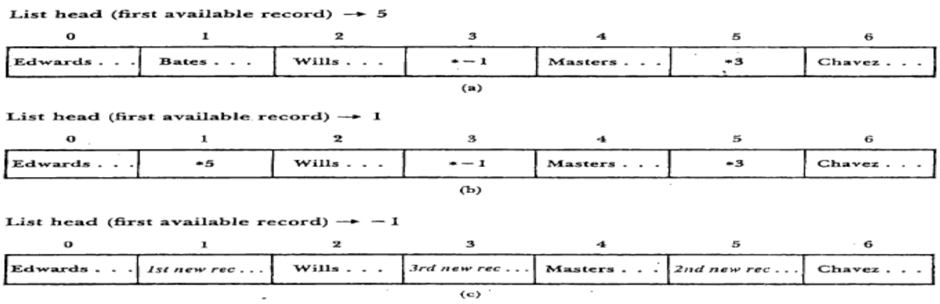


Figure 6.5 Sample file showing linked lists of deleted records. (a) After deletion of records 3 and 5, in that order. (b) After deletion of records 3, 5, and 1, in that order. (c) After insertion of three new records.

When a list is made up of deleted records that have become available space within the file – usually called an avail list.

Stack – We can maintain the list as a stack

A way to know immediately if there are empty slots in the file

A way to jump directly to one of the those slots if it exists.

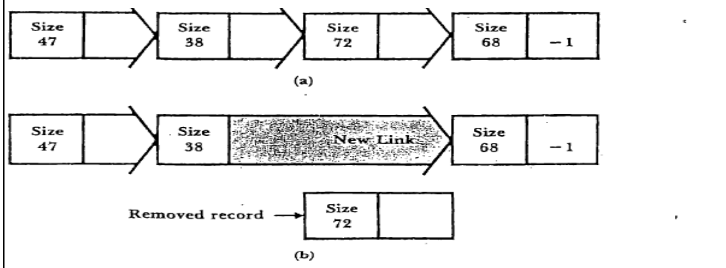
Implementing Fixed-length Record Deletion – When we delete a record, we must be able to mark the record as deleted and then place it on the avail list. A simple way to do this is to place an mark, followed by RRN of the next record on the avail list.

Deleting Variable Length Record

A way to deleted records together into a list.

An algorithm for adding newly deleted records to the avail list and

An algorithm for finding and removing records from avail list where we are ready to use</td><td>10</td><td>CO1</td><td>L1</td></tr>
</table>

	An available List of Variable Length Records What kind of file structure needed? File structure of variable length record, length of each record placing at beginning of each record. We cannot use relative record number to reuse the deleted space.  **Figure 6.7** Removal of a record from an avail list with variable-length records. (a) Before removal. (b) After removal. To identify the correct block size for inserting new record, we have to used placement strategy. First fit placement strategy - We can keep the available list in ascending order based size. Best Fit – takes more time to find out appropriate best memory space Worst Fit – waste more space			
6.	Justify how the compression improves performance? Explain run length encoding for image compression. We should make files smaller, because Use less Storage, resulting in cost savings; Can be transmitted faster, decreasing access time or, alternatively, allowing the same access time with a lower and cheaper bandwidth; and Can be processed faster sequentially. Data Compression involves encoding the information in a file in such a way that it takes up less space. Many different techniques are available for compression. Using a Different Notation Reduce the number of bits for each field in a record to bytes and make the record as compact notation. Eg., to store state in person class, we can use different notation to save space What are the costs of this compression scheme, By using a pure binary encoding, we have made the file understandable by humans	10	CO1	L2

	We incur some cost in encoding time whenever we add a new state-name field to our file and a similar cost for decoding when we need to get a readable version of the state name from the file. Supressing Repeating Sequences This type of compression can be used in images The run-length-encoding technique can be used here to compress image file. The special run-length code indicator; pixel value that is repeated and number of times that the value is repeated Eg., 22 23 24 24 24 24 24 24 24 25 26 26 26 26 26 26 25 24 22 23 ff 24 07 25 ff 26 06 25 24			

Faculty Signature CCI Signature HOD Signature