Unit – 4

Part –A

1) Define the terms – file, file path, directory?

File: A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks.

File path:

Directory :

2) Explain any four common file attributes?

- Read-only - Allows a file to be read, but nothing can be written to the file or changed.
- Archive - Tells Windows Backup to backup the file.
- System - System file.
- Hidden - File will not be shown when doing a regular dir from DOS.

3) Explain any four file operations?

- Copying a file
- Moving a file
- Deleting a file
- Renaming a file

4) Distinguish between shared and exclusive lock?

exclusive	**Shared**
exclusive locks are sometimes called "write locks".	Shared locks are sometimes called "read locks"
An exclusive lock prohibits other users from reading the locked resource	a shared lock allows other users to read the locked resource, but they cannot update it.

5) List any four common file types and their extensions?

AIFF or .AIF	Audio Interchange File Format
.AU	Basic Audio
.AVI	Multimedia Audio/Video
.BAT	PC batch file
.BMP	Windows BitMap
.CLASS or .JAVA	Java files

6) Explain the information associated with an open file?

1 File pointer
2 File open count
3 Disk location of the file
4 Access rights

7) List the different file accessing methods?

- Sequential access
- Direct/Random access
- Indexed sequential access

8) Explain the operations that can be performed on a directory?

- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file
- Traverse the file system

9) Discuss the most common schemes for defining the logical structure of a directory?

The most common schemes for defining the logical structure of a directory
1 Single-Level Directory
2 Two-level Directory
3 Tree-Structured Directories
4 Acyclic-Graph Directories
5 General Graph Directory

10) Describe UFD and MFD.?

In the two-level directory structure, each user has her own user file directory (UFD). EachUFD has a similar structure, but lists only the files of a single user. When a job starts thesystem's master file directory (MFD) is searched. The MFD is indexed by the user name oraccount number, and each entry points to the UFD for that user.

11) Describe file system mounting?

In computers, to mount is to make a group of files in a file system structure accessible to a user or user group. In some usages, it means to make a device physically accessible. For instance, in data storage, to mount is to place a data medium (such as a tape cartridge) on a drive in a position to operate. Macintosh calls it mounting when a user inserts a disc into the machine.

12) Write the format of a typical file-control block?

A File Control Block (FCB) is a file system structure in which the state of an open file is maintained. A FCB is managed by the operating system, but it resides in the memory of the program that uses the file, not in operating system memory. This allows a process to have as many files open at one time as it wants to, provided it can spare enough memory for an FCB per file**.**

13) List the different disk-space allocation methods?

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

14) List the various layers of a file system?

- Application programs
- Logical file system
- File-organization module
- Basic file system
- I/O control
- Devices

15) Explain the functions of virtual file system (VFS)?

It has two functions

i. It separates file-system-generic operations from their implementation defining a clean VFS interface. It allows transparent access to different types of file systems mounted locally.

ii.VFS is based on a file representation structure, called a vnode. It contains a numerical value for a network-wide unique file .The kernel maintains one vnode structure for each active file or directory.

16) Describe about different types of disk scheduling?

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.

SSTF: In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first.
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.
CSCAN: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction.
LOOK: 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.

CLOOK: As LOOK is similar to SCAN algorithm, in similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm inspite 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.

17) Define the terms with respect to disk I/O - seek time, latency time?

Seek time is the time required to move the disk arm to the required track. latency is the time it takes for the beginning of the required sector to reach the head.

18) Explain the allocation methods of a disk space?

The allocation methods define how the files are stored in the disk blocks. There are three main disk space or file allocation methods.

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

The main idea behind these methods is to provide:

- Efficient disk space utilization.
- Fast access to the file blocks.

19) State the advantages of linked disk-space allocation strategy?

Advantages:

- This is very flexible in terms of file size. File size can be increased easily since the system does not have to look for a contiguous chunk of memory.
- This method does not suffer from external fragmentation. This makes it relatively better in terms of memory utilization

(or)

- No external fragmentation
- Size of the file does not need to be declared

20) State the advantages of indexed disk-space allocation strategy?

Advantages:

- This supports direct access to the blocks occupied by the file and therefore provides fast access to the file blocks.
- It overcomes the problem of external fragmentation.

(or)

- No external-fragmentation problem
- Solves the size-declaration problems.
- Supports direct access

21) List the different free disk-space management techniques?

- Bitmap
- Linked Lists
- Grouping
- Counting

22) Explain the bit vector method free space management on disk?

The free-space list is often implemented as a bit map or a bit vector

If a block is free, the bit is 1

If a block is allocated, the bit is 0.

23) Discuss the advantages of contiguous memory allocation of disk space?

- Both the Sequential and Direct Accesses are supported by this. For direct access, the address of the kth block of the file which starts at block b can easily be obtained as (b+k).
- This is extremely fast since the number of seeks are minimal because of contiguous allocation of file blocks.

(Or)

- Supports direct access
- Supports sequential access
- Number of disk seeks is minimal.

24) Discuss the drawbacks of contiguous allocation of disk space?

- This method 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**.**

(or)

- Suffers from external fragmentation
- Suffers from internal fragmentation
- Difficulty in finding space for a new file
- File cannot be extended
- Size of the file is to be declared in advance

25) List any four secondary storage memory devices?

- USB drives
- Floppy disks
- CD-R
- DVD-R
- Magnetic tapes

26) Describe about logical formatting of the disk?

27) List various disk-scheduling algorithms?

- First Come-First Serve (FCFS)
- Shortest Seek Time First (SSTF)
- Elevator (SCAN)
- Circular SCAN (C-SCAN)
- LOOK
- C-LOOK

28) State the purpose of boot block?

Boot Block: located in the first few sectors of a file system. The boot block contains the initial bootstrap program used to load the operating system. Typically, the first sector contains a bootstrap program that reads in a larger bootstrap program from the next few sectors, and so forth.

Part-B

1)

a) Discuss the criteria for choosing a fileorganization?

A)a)The criteria are:

1)Fast access to single record or collection of related records.

2)Easy record adding/update/removal without disrupting.

3)Storage efficiency.

4)Redundancy as a warranty against data corruption.

b) Describe indexed file and indexed sequential fileorganization?

b)**Indexed file organization:**

An indexed file contains records ordered by a *record key*. A record key uniquely identifies a record and determines the sequence in which it is accessed with respect to other records.

Each record contains a field that contains the record key. A record key for a record might be, for example, an employee number or an invoice number.

An indexed file can also use *alternate indexes*, that is, record keys that let you access the file using a different logical arrangement of the records. For example, you could access a file through employee department rather than through employee number.

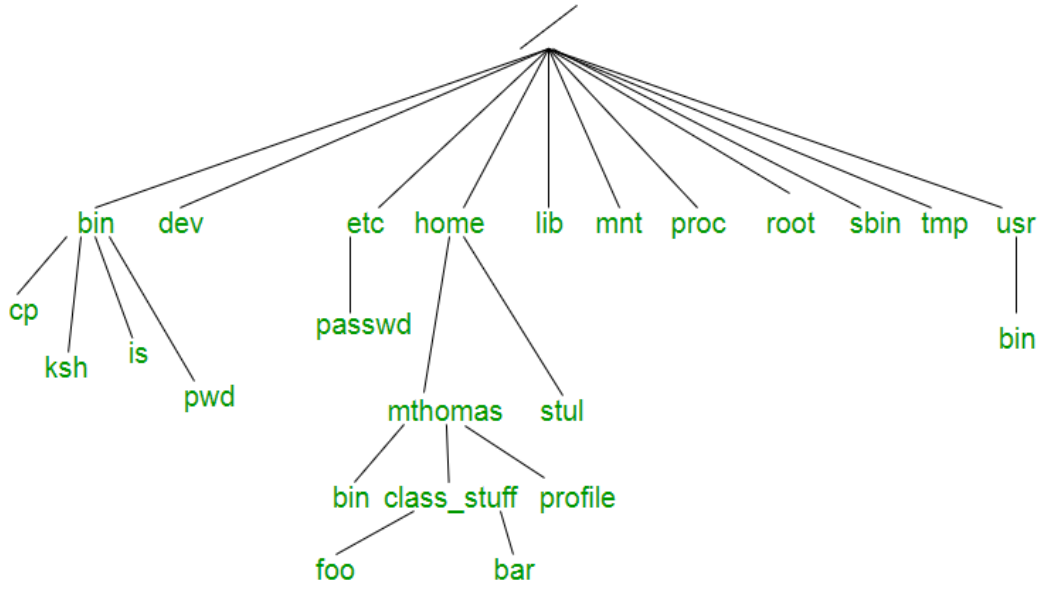
The possible record transmission (access) modes for indexed files are sequential, random, or dynamic. When indexed files are read or written sequentially, the sequence is that of the key values.

Indexed-sequential file organization method:

Almost similar to sequential method only that, an index is used to enable the computer to locate individual records on the storage media. For example, on a magnetic drum, records are stored sequential on the tracks. However, each record is assigned an index that can be used to access it directly.

2) Describe the file system of UNIX?

- Unix file system is a logical method of **organizing and storing** large amount of information in a way which makes it easy to manage.
- The file is the smallest unit in which information is stored.
- The Unix file system has several important features. All data in Unix is organized into files.All files are organized into directories. These directories are organized into a tree-like structure called the file system.
- Files in Unix System organized in multi-level hierarchy structure called a directory tree. At the very top of the file system is a directory called “root” which is represented by a “/”. All other files are “descendants” of root.



4 Differentiate among the following disk scheduling algorithms? a) FCFS b) SSTF c) SCAN d) C-SCAN e) LOOK f) C-LOOK

A)

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 optimize seek time
- May not provide the best possible service

2. SSTF: In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first. So, the seek time of every request is calculated in advance in 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 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

3. 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 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

4. 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 *CSAN* 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 SCAN algorithm and hence it is known as C-SCAN (Circular SCAN).

Advantages:

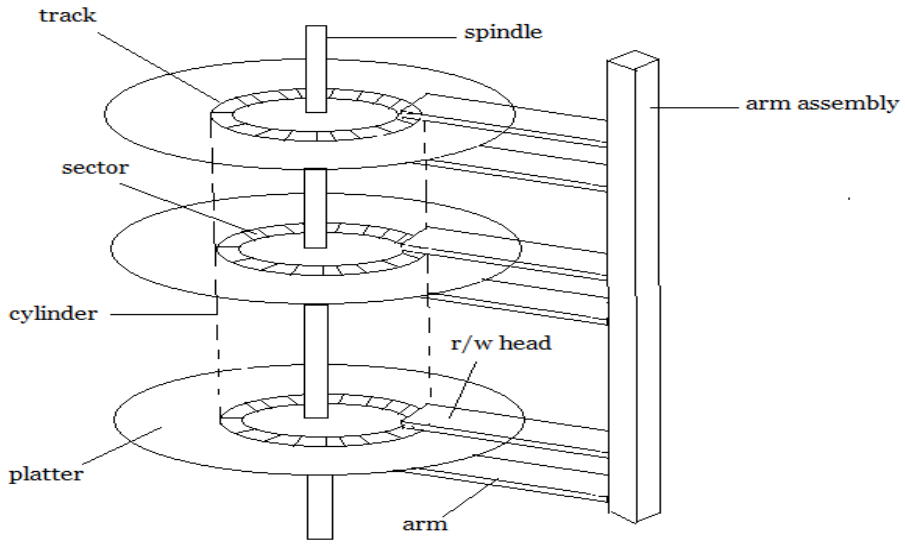
- Provides more uniform wait time compared to SCAN

5. LOOK: 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.
6. CLOOK: As LOOK is similar to SCAN algorithm, in similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm inspite 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.

5)a)Explain magnetic disk structure and its management?

b) Exemplify swap space management?

A) a)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 centre is less than the length of the tracks farther from the centre. Each track is further divided into sectors, as shown in the figure.

Tracks of the same distance from centre 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.

b) 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*.

6) Explain the following in detail with respect to disk? a) Seek time b) Latency c) Accesstime d) Transfer time

A) Seek time is the time required to move the disk arm to the required track. Rotational delay or latency is the time it takes for the beginning of the required sector to reach the head. Sum of seek time (if any) and latency is the access time. Time taken to actually transfer a span of data is transfer time.

7) a) Explain in detail the interrupts and interrupt handlingfeatures?

b) Explain with neat diagram the steps in DMA transfer?

A) Interrupts: asynchronous interrupts generated by hardware.

An interrupt handler or interrupt service routine (ISR) 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 part of the device's driver (the kernel code that manages the device).

In Linux, interrupt handlers are normal C functions, which match a specific prototype and thus enables the kernel to pass the handler information in a standard way. What differentiates interrupt handlers 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. This special context is occasionally called atomic context because code executing in this context is unable to block.

Because 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 operating system services the interrupt without delay.
- To the rest of the system: the interrupt handler executes in as short a period as possible.

At the very least, an interrupt handler's job is to acknowledge the interrupt's receipt to the hardware. However, interrupt handlers can oftern have a large amount of work to perform.

b)

- For devices that transfer large quantities of data (such as disk controllers), it is wasteful to tie up the CPU transferring data in and out of registers one byte at a time.
- Instead this work can be off-loaded to a special processor, known as the *Direct Memory Access, DMA, Controller.*
- The host issues a command to the DMA controller, indicating the location where the data is located, the location where the data is to be transferred to, and the number of bytes of data to transfer. The DMA controller handles the data transfer, and then interrupts the CPU when the transfer is complete.
- A simple DMA controller is a standard component in modern PCs, and many *bus-mastering* I/O cards contain their own DMA hardware.
- Handshaking between DMA controllers and their devices is accomplished through two wires called the DMA-request and DMA-acknowledge wires.
- While the DMA transfer is going on the CPU does not have access to the PCI bus (including main memory), but it does have access to its internal registers and primary and secondary caches.
- DMA can be done in terms of either physical addresses or virtual addresses that are mapped to physical addresses. The latter approach is known as *Direct Virtual Memory Access, DVMA,* and allows direct data transfer from one memory-mapped device to another without using the main memory chips.
- Direct DMA access by user processes can speed up operations, but is generally forbidden by modern systems for security and protection reasons. (I.e. DMA is a kernel-mode operation.)

- Figure 13.5 below illustrates the DMA process.

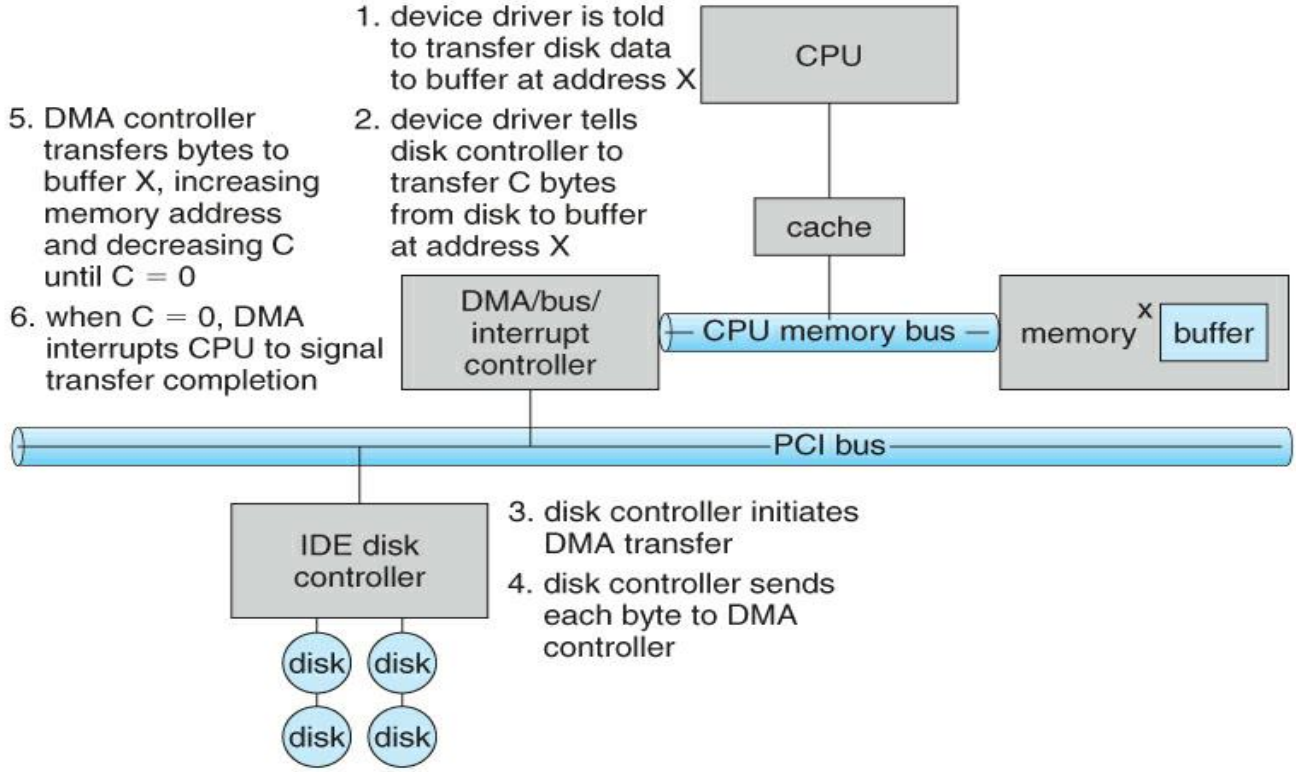


Figure 13.5 - Steps in a DMA transfer.

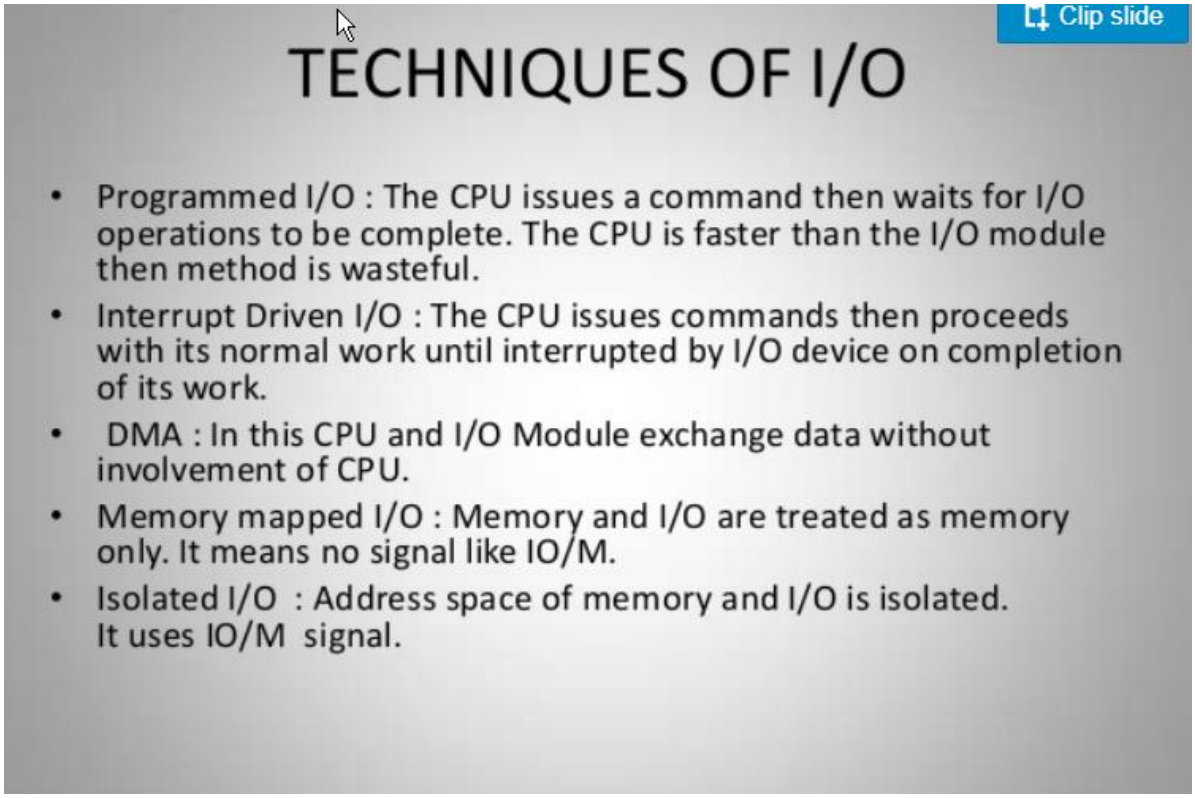
8) a) Discuss the N-step SCAN policy for disk scheduling?

b) Explain how double buffering improves the performance than a single buffer for I/O?

A) a) N-step SCAN: like SCAN, except, disk request queue segmented into sub queues of fixed-length N. Sub queues are processed one at a time using SCAN inside each sub queue. When N is large, it approaches SCAN. When N=1, it approaches FCFS. Problem: does not adjust N for varying loads. When the load of requests increases, we would like N to increase.

b) A process transfers data to (or from) one buffer while the operating system empties (or fills) the other. Double buffering ensures that the process will not have to wait on I/O

9) a) Explain the techniques used for performing I/O?



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

A) a)

b) a. Print the content of the file. b. Print the content of record i. This record can be found using hashing or index techniques

10) Discuss in detail the performance issues of secondary storage management?

A)

- The speed of the secondary storage is also lesser than that of primary storage.
- Hence, the data which is less frequently accessed is kept in the secondary storage.
- A few examples are magnetic disks, magnetic tapes, removable thumb drives etc.

- Secondary storage devices do not interact directly with an application.
- The purpose of secondary storage is to provide a high-capacity tier, although the data stored is not immediately accessible. For example, a backup server is capable of storing a vast amount of data, but getting access to it requires dedicated backup software.

11) Explain how disk caching can improve disk performance?

A) 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.

12) Explain low-level formatting or physical formatting?

- A) Low-level formatting is the process of marking out cylinders and tracks for a blank hard disk, and then dividing tracks into multiple sectors.
- This process is often called the "*real*" formatting since it creates physical format which defines where the data is saved.
- If users perform low-level formatting when data have been installed, all existing files will be erased, and it is almost impossible to recover them. Therefore, some users make such a format to avoid privacy leakage.
- Nevertheless, performing low-level formatting will bring great influence on hard disk, thus shortening hard disk service time. Therefore, it is not suggested.
- To low level format a hard disk, users can make use of specific tools as well as Debug assembler languages. That's relatively complicated.

13) Define buffering, caching and spooling?

A) Spooling:

- Acronym of "Simultaneous Peripheral Operation On-Line".
- Its a process of placing data in 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), there 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 process can access the resource without waiting.
- After writing the data on spool, process can perform other tasks. And printing process operates seperately.
- Without spooling, process would be tied up until the printing finished.
- Spooling is useful for the devices which have differing data access rate. Used mainly when processes share some resource and needed to have synchronization.

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.
- Buffer may be used in between when moving data between two processes within a computer. Data is stored in buffer as it is retrieved from one processes 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.

Caching:

- Caching transparently stores data in component called Cache, so that future request 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 devi ce.
- 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.
- A cache's sole purpose is to reduce accesses to the underlying slower storage.

14) Discuss the following a) File system mounting b) Thrashing

A)

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.

15) Explain the following file concepts: a) File attributes b) File operations c) File types d) Internal file structure

A) 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

File Operations

- The 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 OSes 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.
    - Disk location of the file.
    - Access rights
- Some systems provide support for *file locking*.
    - A *shared lock* is for reading only.
    - A *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.)
    - UNIX used advisory locks, and Windows uses mandatory locks.

File Types

- Windows (and some other systems) use special file extensions to indicate the type of each file:
- Macintosh stores a creator attribute for each file, according to the program that first created it with the create( ) system call.
- UNIX stores magic numbers at the beginning of certain files. (Experiment with the "file" command, especially in directories such as /bin and /dev)

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 organized in units of 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.

16) Explain the concept of file sharing? What are the criteria to be followed in systems which implement file sharing?

A)

17) Describe the following Directory Implementation methods? a) Linear List b) Hash Table6

A) 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

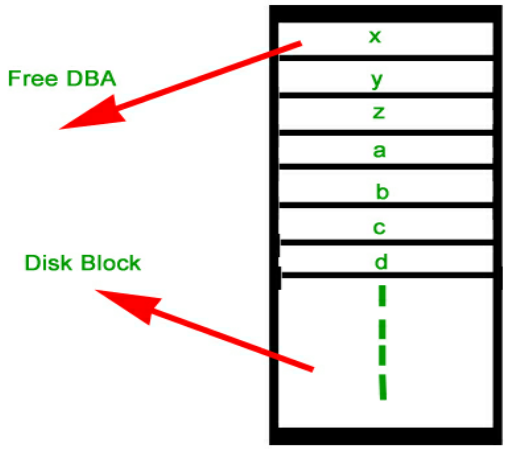
18) Explain the concept and techniques of free space management?

A)Just as the space that is allocated to files must be managed ,so the space that is not currently allocated to any file must be managed. To perform any of the file allocation techniques,it is necessary to know what blocks on the disk are available. Thus we need a disk allocation table in addition to a file allocation table.The following are the approaches used for free space management.

1. Bit Tables : This method uses a vector containing one bit for each block on the disk. Each entry for a 0 corresponds to a free block and each 1 corresponds to a block in use.
   For example: 00011010111100110001

In this vector every bit correspond to a particular vector and 0 implies that, that particular block is free and 1 implies that the block is already occupied. A bit table has the advantage that it is relatively easy to find one or a contiguous group of free blocks. Thus, a bit table works well with any of the file allocation methods. Another advantage is that it is as small as possible.

2. Free Block List : In this method, each block is assigned a number sequentially and the list of the numbers of all free blocks is maintained in a reserved block of the disk.



A) 19) Discuss about a) Disk space management b) Swap -space management

A) a)

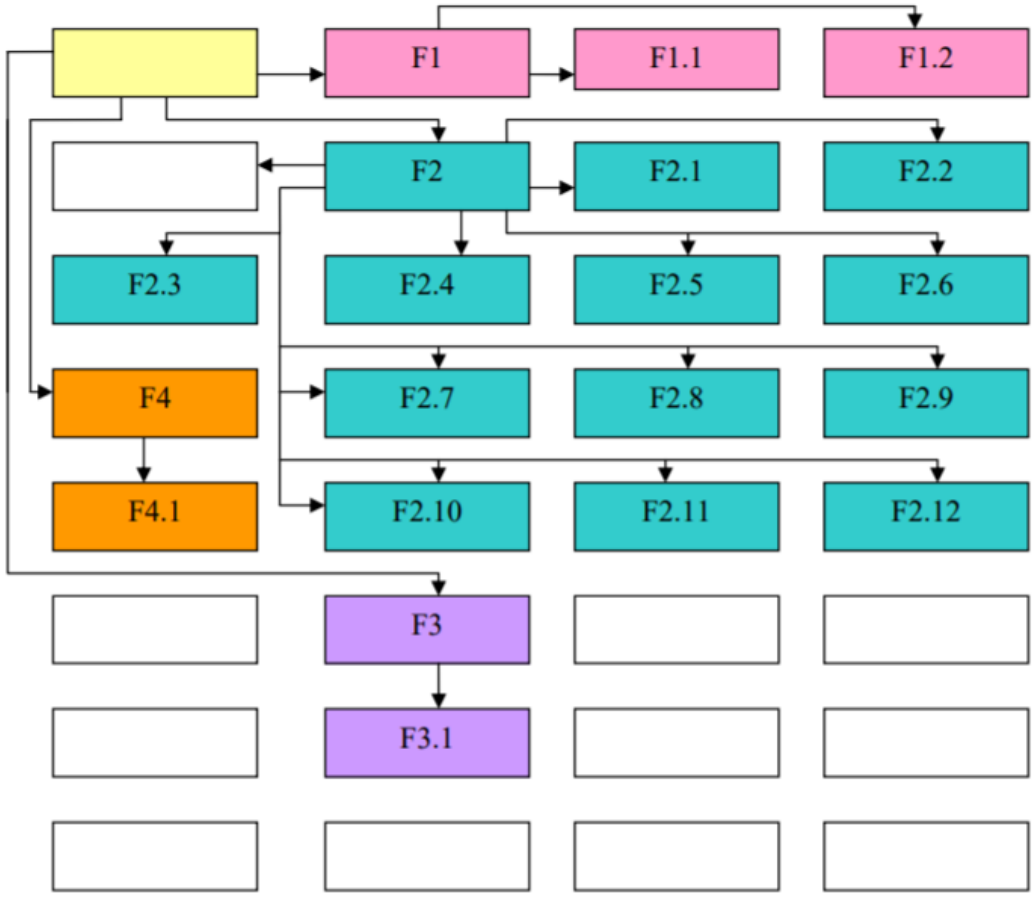
b)refer part b 5)b)

PART-C

2) Using a diagram, show how an indexed allocation of a file may be done for a disked based system with the following characteristics. The disc size is 30blocks each of 1024 bytes (may be modeled as 6 X 5 matrixes). File f1 is 11 logical records of 112 bytes, file f2 is 890 logical records of 13 bytes, file f3 is 510 bytes of binary data stream and file f4 is 4 logical blocks of 95 bytes.

Ans: See the figure below.

Here, each block is of 1024 bytes.



- File f1 is 11 logical records each of 112 bytes; that means a total of 11 * 112 bytes = 1232 bytes. This implies it needs (1232 / 1024) ~ 2 blocks
- File f2 is 890 logical records each of 13 bytes; that means a total of 13 * 890 bytes = 11570 bytes. This implies it needs (11570 / 1024) ~ 12 blocks.
- File f3 is 570 bytes of binary data stream; that means it needs 1 block.
- File f4 is 4 blocks each of 95 bytes; that means a total of 4 * 95 bytes = 380 bytes.

This implies it needs 1 block.

4)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
= 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 gigs

5) Discuss the reasons why the operating system might require accurate information on how blocks are stored on a disk. How could the operating system improve file system performance with this knowledge?

Answer: 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.

6) Discuss how an operating system could maintain a free-space list for a tape-resident file system. Assume that the tape technology is append-only, and that it uses the EOT mark and locate, space, and read position commands

Answer:

- 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.

7) Is there any way to implement truly stable storage? Explain your answer.

Answer: 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.

8) Could a RAID Level 1 organization achieve better performance for read requests than a RAID Level 0 organization (with non redundant striping of data)? If so , how?

Answer: Yes, a RAID Level 1 organization could achieve better performance for read requests. When a read operation is performed, a RAID Level 1 system can decide which of the two copies of the block should be accessed to satisfy the request. This choice could be based on the current location of the disk head and could therefore result in performance optimizations by choosing a disk head that is closer to the target data

9)Compare the performance of write operations achieved by aRAIDLevel5 organization with that achieved by a RAID Level 1organization.

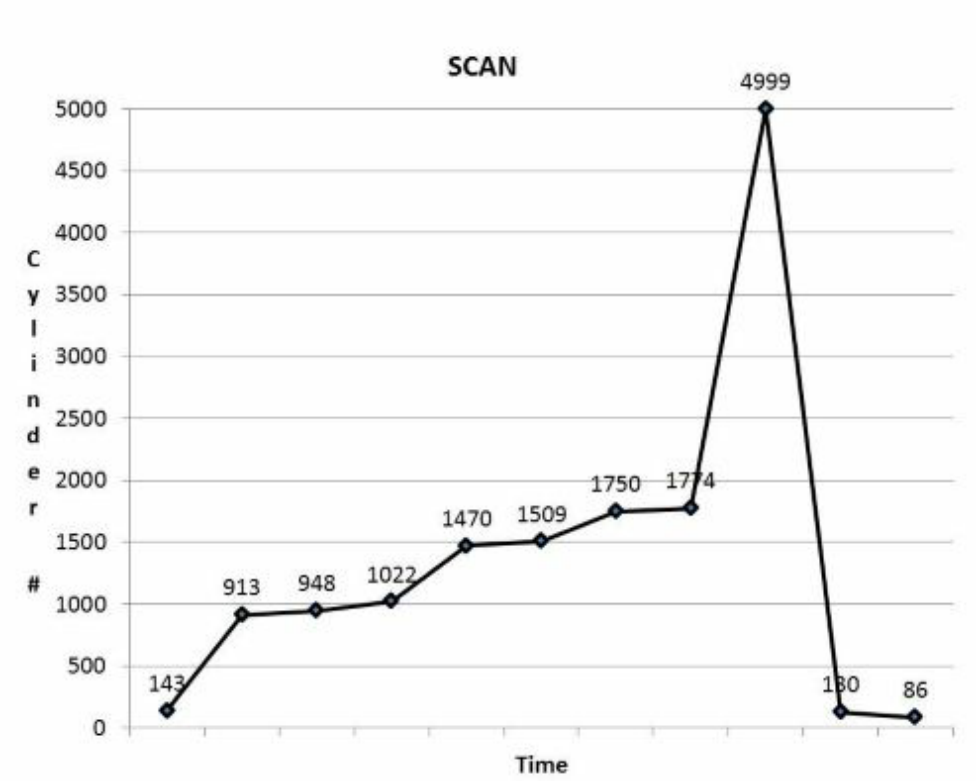
Answer: RAID Level 1 organization can perform writes by simply issuing the writes to mirrored data concurrently. RAID Level 5, on the other hand, would require the old contents of the parity block to be read before it is updated based on the new contents of the target block. This results in more overhead for the write operations on a RAID Level5 system

10)Consider that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving request 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.



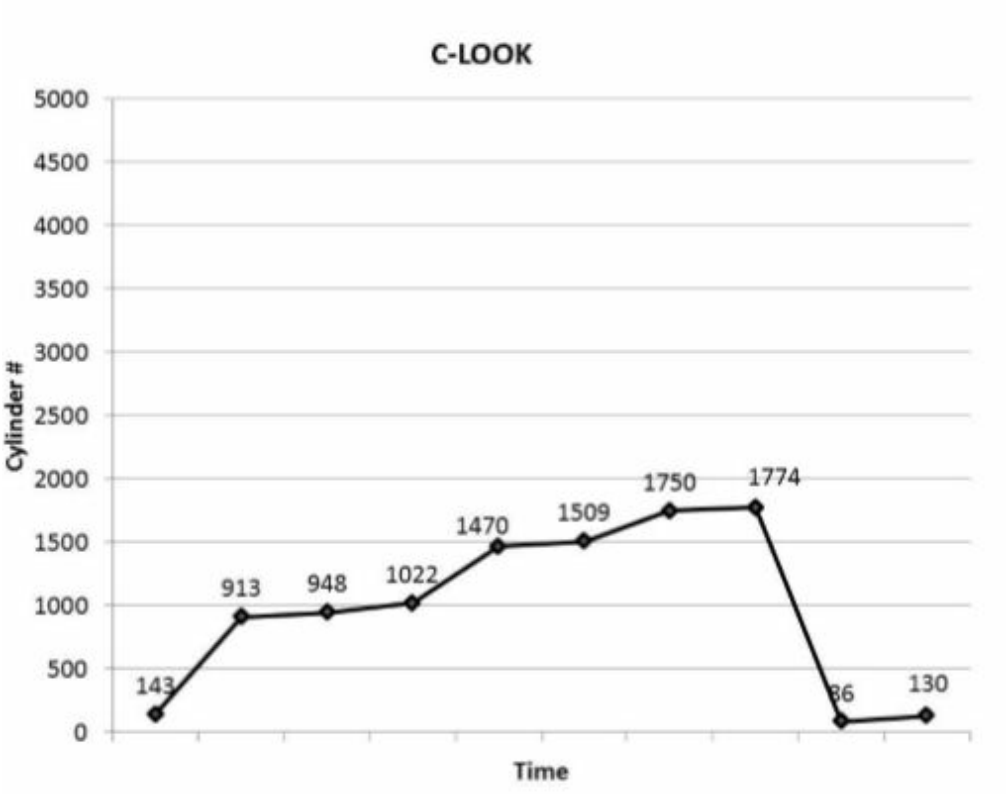
Total Distance travelled = |143−913|+|913−948|+|948−1022|+|1022−1470|+|1470−1509|+|1509−1750|+|1750−1774|+|1774−4999|+|4999−180|+|180−86|=9769

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|+|1750−1774|+|1774−86|+|86−130|=3363



Total distance travelled= |143−913|+|913−948|+|948−1022|+|1022−1470|+|1470−1509|+|1509−1750|+|1750−1774|+|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.