UNIT IV
I/O SYSTEMS

MASS-STORAGE SYSTEMS

APEC

OVERVIEW OF MASS STORAGE STRUCTURE

- Magnetic Disks
- Magnetic tapes

Secondary storage Devices

APEC

MAGNETIC DISKS

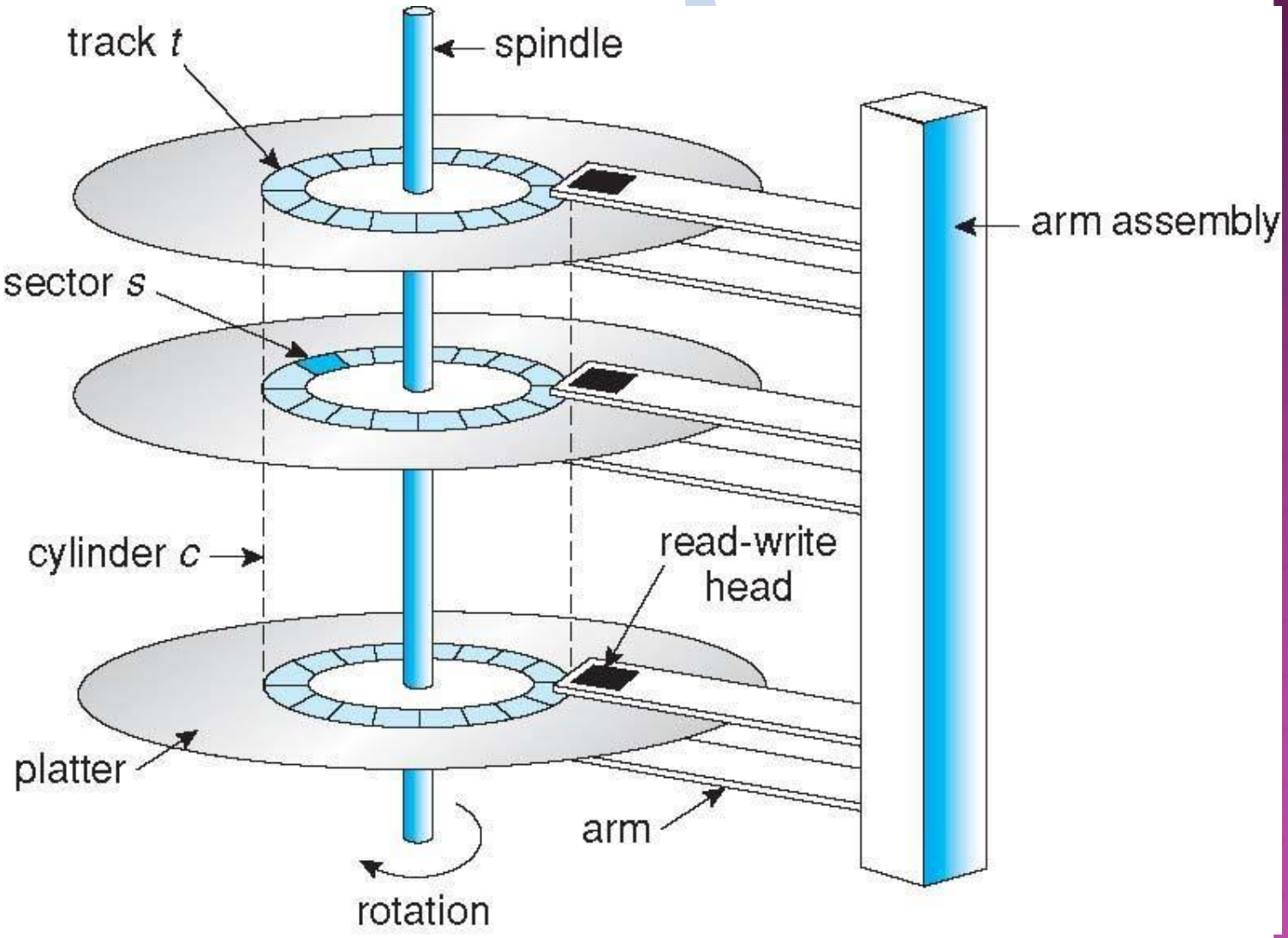
- Disk ➔ 1.8 to 5.25 inches
- Magnetic disk consists of Plate/Platter, which is made up of glass material.
  - If data is stored on one side ➔ single sided disk
  - If data is stored on double sides ➔ double sided disk
- If disk surface of the platter is further divided into **tracks** and **sectors**.
- **Tracks** ➔ concentric circles, where data is stored from outermost to innermost.
  - (ie) 50,000 to 100,000 tracks/Platter
- Hard disk have 1 to 5 Platters.
- Tracks are further divided into sectors. Sector is like an arc. It is

APEC

the smallest unit of information that is transferred to/from the disk.

APEC

MOVING-HEAD DISK MECHANISM



CONTD...

- Sector size ➔ 512 bytes.
  - 500 sectors/track in the inner track
  - 1000 sectors/track in the outer track

- Disk contains read/write head for each surface of the platter. All read/write heads are attached to assembly disk arm.
- The disk platters ➔ mounted on spindle together. The track of equal diameter on different platters form a cylinder.
- The disk sometimes removable, floppy disks ➔ inexpensive.

APEC

MAGNETIC TAPES

- It hold large quantities of data, but its slower than main memory and magnetic disk.
  - Tapes ➔ store from 20 GB to 200 GB

APEC

DISK SCHEDULING

- The operating system is responsible for having a fast access time and large bandwidth

Access time = Seek time + Rotational Latency

  - **Seek time:** Time taken for the disk are to move the heads containing the desired sector
  - **Rotational latency:** Time taken for the disk to rotate the desired sector to the disk head

- $$Disk\ bandwidth = \frac{Total\ number\ of\ bytes\ transferred}{\quad}$$

Time taken between the request & completion of service

APEC

DISK SCHEDULING TYPES

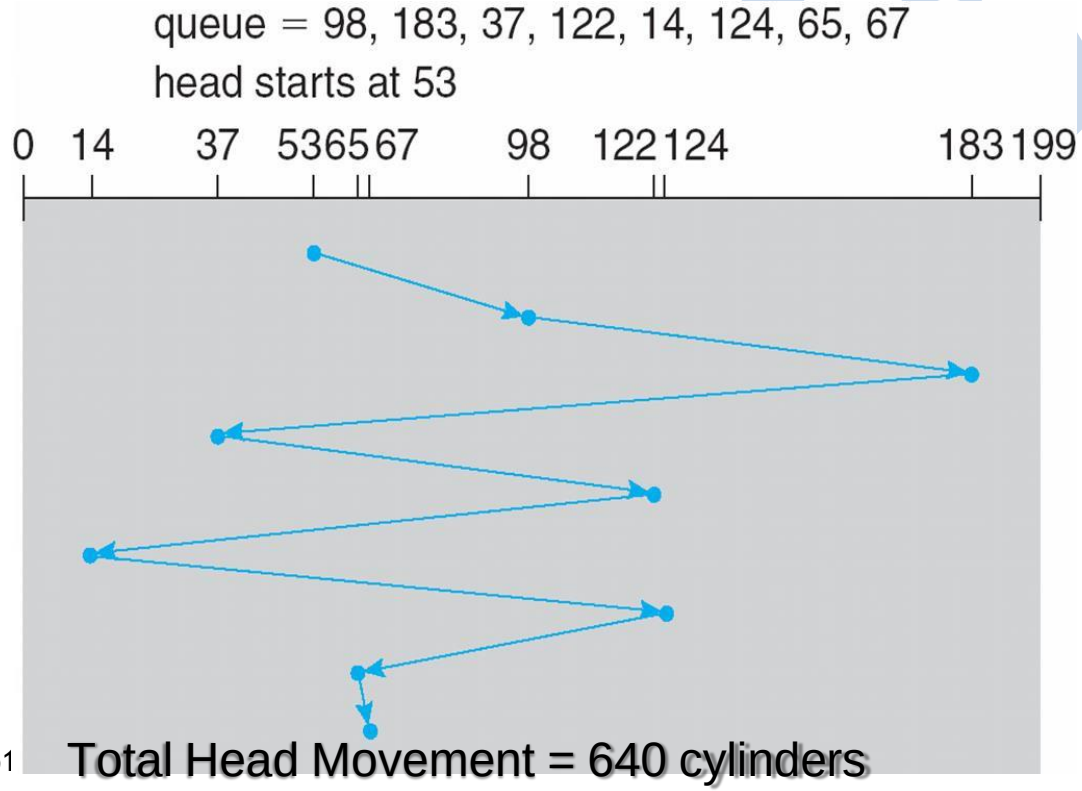
- FCFS scheduling
- SSTF scheduling
- SCAN scheduling
- C – SCAN scheduling
- LOOK scheduling

APEC

FCFS SCHEDULING

- First come first serve scheduling

Example: disk Queue with requests



Total Head Movement = 640 cylinders

53 – 98 = 45

98 – 183 = 85

183 – 37 = 146

37 - 122 = 85

122 -14 = 108

14 – 124 = 110

124 – 65 = 59

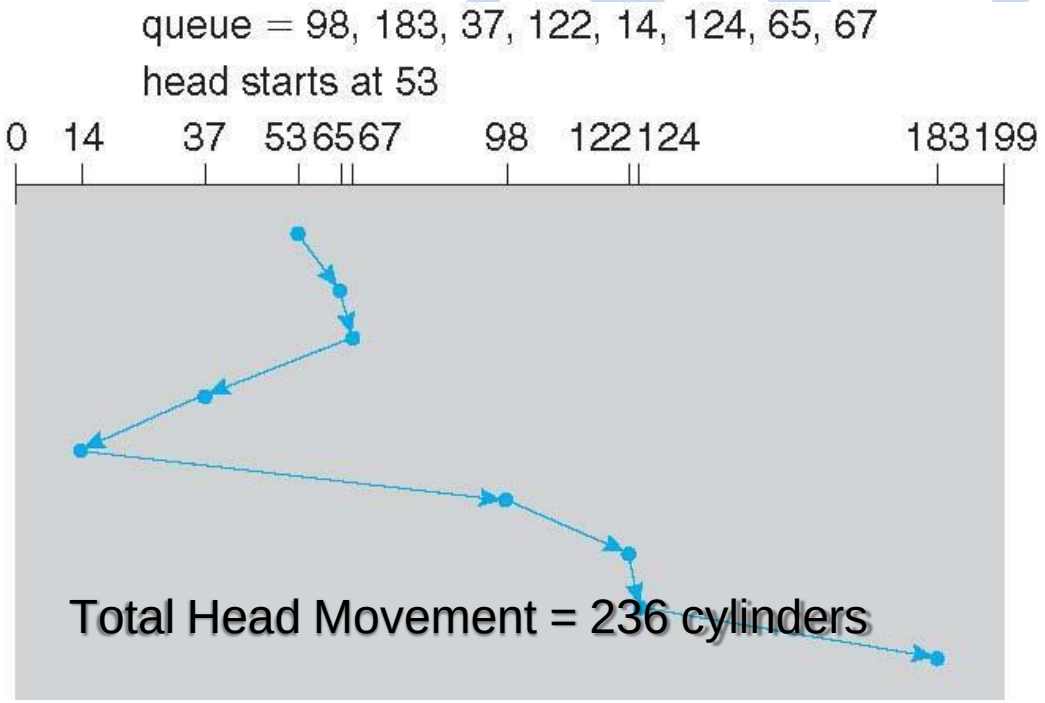
65 – 67 = 2

Total = 640

SSTF SCHEDULING

SHORTEST SEEK TIME FIRST SCHEDULING

- Service all the requests closest to the current head position.



53 - 65 = 12

65 – 67 = 2

67 – 37 = 30

37 – 14 = 23

14 – 98 = 84

98 – 122 = 24

22 – 124 = 2

124 – 183 = 59

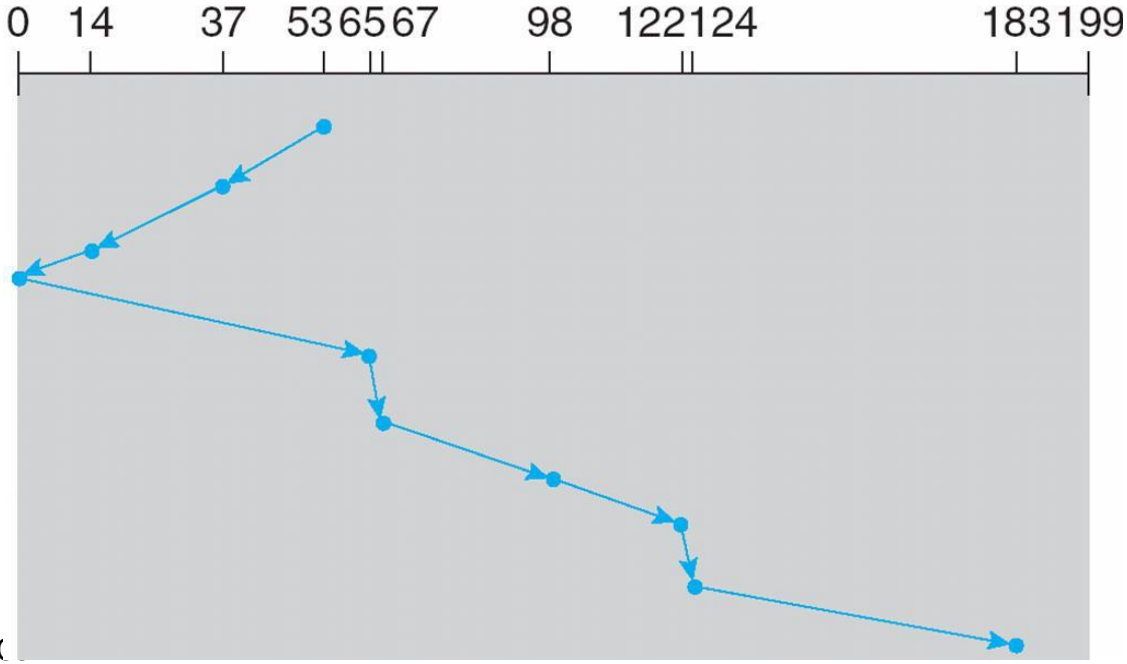
Total = 236

SCAN SCHEDULING

- The disk arm starts at one end of the disk, and moves toward the other end, servicing all the requests of the cylinder.
- **SCAN algorithm** Sometimes called the **Elevator Algorithm**

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



Total Head Movement = 208 cylinders

53 - 37 = 16

37 – 14 = 23

14 – 65 = 51

65 – 67 = 2

67 – 98 = 31

98 – 122 = 24

122 – 124 = 2

124 – 183 = 59

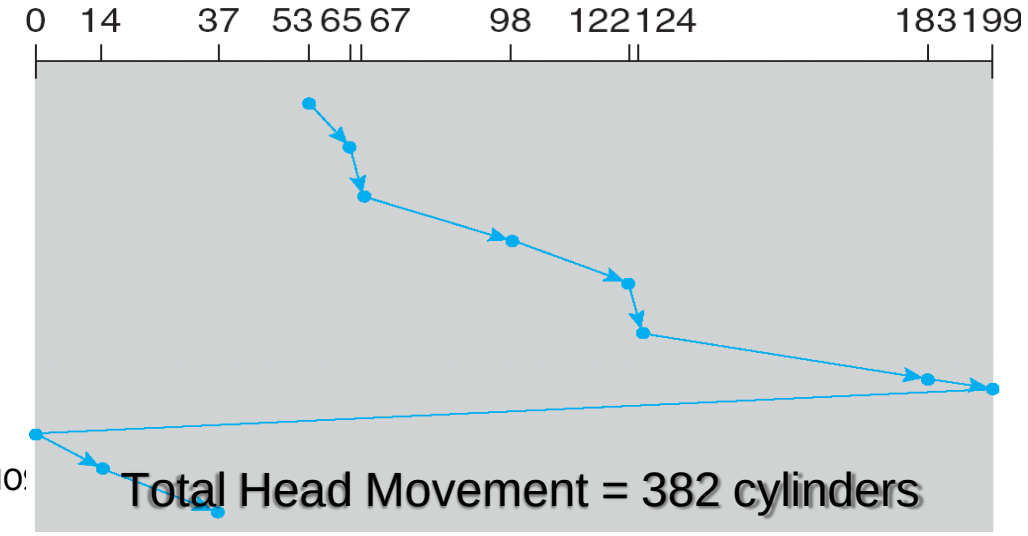
Total = 208

C-SCAN SCHEDULING

- Variant of SCAN scheduling used to provide a more uniform wait time.
- C-SCAN, moves the head from one end of the disk to the other, servicing requests along the way.
  - When the head reaches the other end, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



53 - 65 = 12

65 – 67 = 2

67 – 98 = 31

98 – 122 = 24

122 – 124 = 2

124 – 183 = 59

183 – 199 = 16

199 – 0 = 199

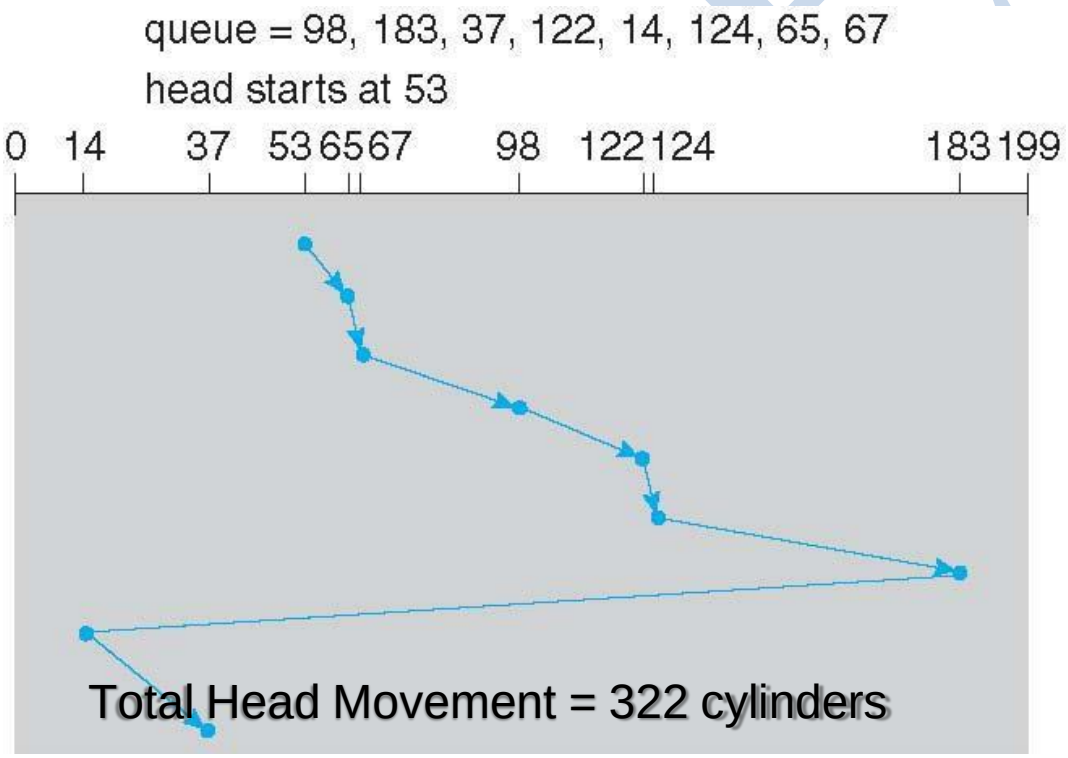
0 – 14 = 14

14 – 37 = 23

Total = 382

LOOK SCHEDULING

- Both SCAN & C-SCA, move the disk arm across the full width of the disk.
- In LOOK scheduling, the arm goes only till the final request.



53 - 65 = 12

65 – 67 = 2

67 – 98 = 31

98 – 122 = 24

122 – 124 = 2

124 – 183 = 59

183 – 14 = 169

14 – 37 = 23

Total = 322

SELECTING A DISK-SCHEDULING ALGORITHM

- When there is heavy load, FCFS, SCAN, C-SCAN performs better.
- Optimal waiting time is achieved in SSTF & SCAN.

APEC

DISK MANAGEMENT

- Disk formatting
- Boot block
- Bad blocks

APEC

DISK FORMATTING

- The Magnetic Disk is like a slate, the disk can store the data and its divided into sectors. This process is called as **Low-level formatting**, or **physical formatting.** It fills the data with the special data structure for each sector.
- The data structure for the sector consists of
  - Header
  - Data area (512 bytes in size)
  - Trailer
- Header + Trailer ➔ contain information used by disk controller, such as sector number and Error Correcting Code (ECC)
- This physical partitioning helps the manufacturer to
  - Test the disk
  - Initialize the mapping from logical block number
- **Major steps:**

APEC

- **Partition** the disk into one or more groups of cylinders
  - One partition hold Oss Executable files while other partitions holds user files
- **2nd step is Logical formatting.** The operating system store the initial data structures in the disk.

APEC

BOOT BLOCK

- Whenever the computer is powered up, it is rebooted and it needs the initial program to run. This initial program is called **Bootstrap Program.**
- This Bootstrap program initialize CPU registers, main contents of memory etc.

STEPS:

- Find the kernel on Disk
- Load kernel into memory
- Jump to the initial address for Execution

- The full bootstrap program is stored in the partition called **boot blocks.**

APEC

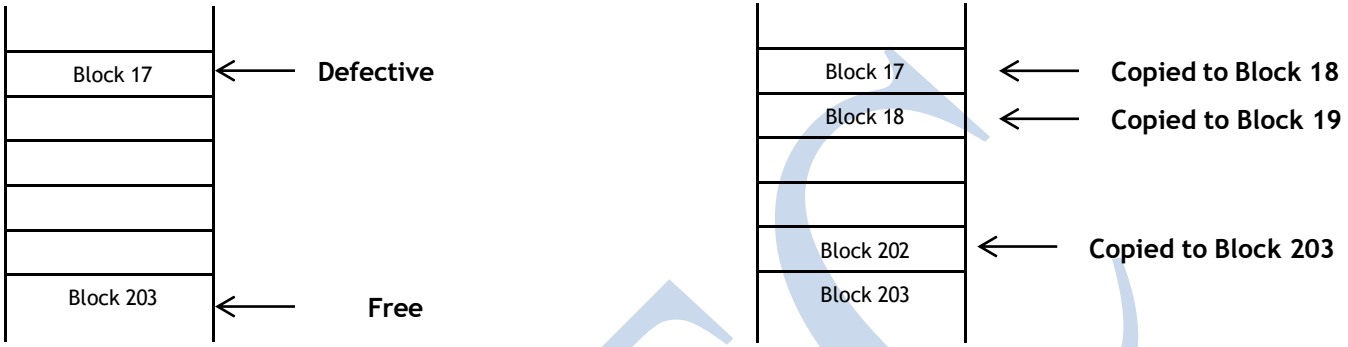
- Bootstrap Loader ➔ load the entire OS from the non-fixed position onto the disk.

APEC

BAD BLOCKS

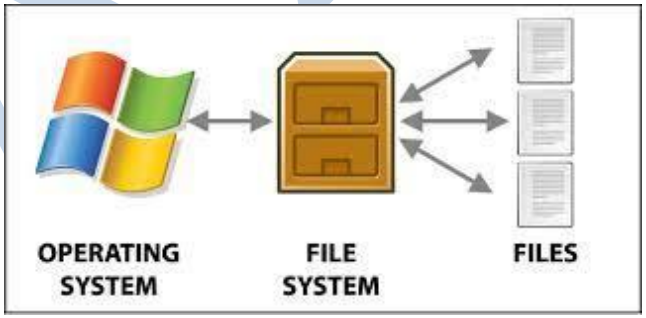
- Disk with defected sector is called bad blocks. Depending on the use of disks, the blocks are handled in 3 ways:
- Method 1: "Handled Manually"
  - If blocks go bad, a special program must run manually to search the bad blocks. Data resided in bad blocks are lost.
- Method 2: "Sector sparing or Forwarding"
  - The controller maintains a list of bad blocks on the disk. The controller replace each bad sector with any one of the spare sectors.
- Method 3: "Sector Slipping"
  - Suppose that logical block 17 becomes defective and the first available spare follows sector 202.

AREC


Block 17
Defective
Block 203
Free
Block 17
Copied to Block 18
Block 18
Copied to Block 19
Block 202
Copied to Block 203
Block 203

APEC

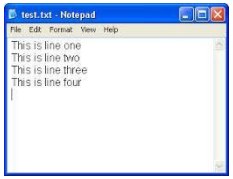
CHAPTER 11: FILE SYSTEM IMPLEMENTATION



FILE SYSTEM

- File:
  - It is abstract data type.
  - Collection of related information (ie) recorded on secondary storage.
- Data cannot be stored on the secondary storage unless its stored as a file.

Examples:



- Text file
  - Sequence of characters organized into lines.
- Source file
  - Executable statements along with the functions
- Object file

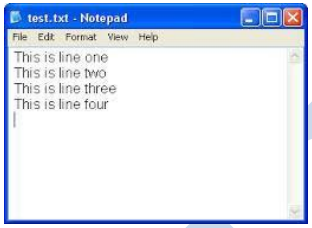
APEC

- Sequence of bytes organized into blocks (ie) understandable by system's linker.

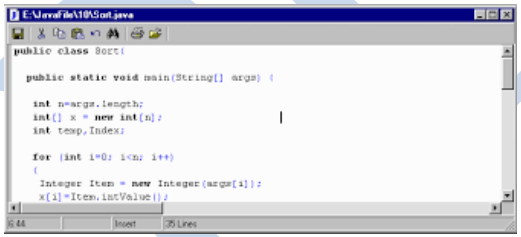
APEC

EXAMPLES OF FILE SYSTEM

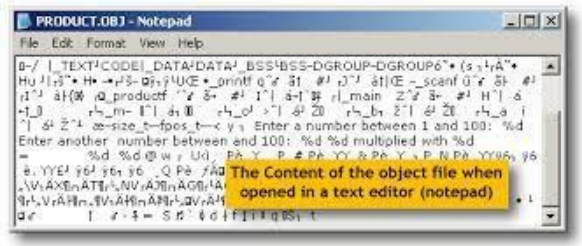
- **Text file**
  - Sequence of characters organized into lines.



- **Source file**
  - Executable statements along with the functions



- **Object file**

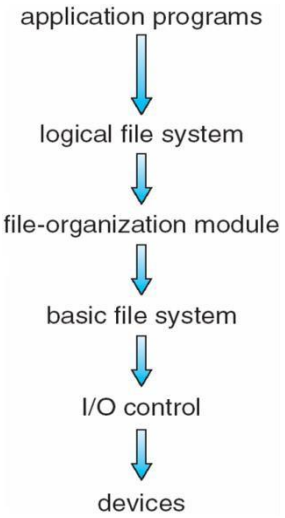


- Sequence of bytes organized into blocks (ie) understandable by system's linker.

APEC

LAYERED FILE SYSTEM

APEC



Application Program

↓ Symbolic Name or file descriptor index (handle)

Logical file system - deals with directory and protection

↓ logical block number, type of file, device, file location

File-Organization module - based on file type translates logical block # to physical block #

↓ physical block number, device

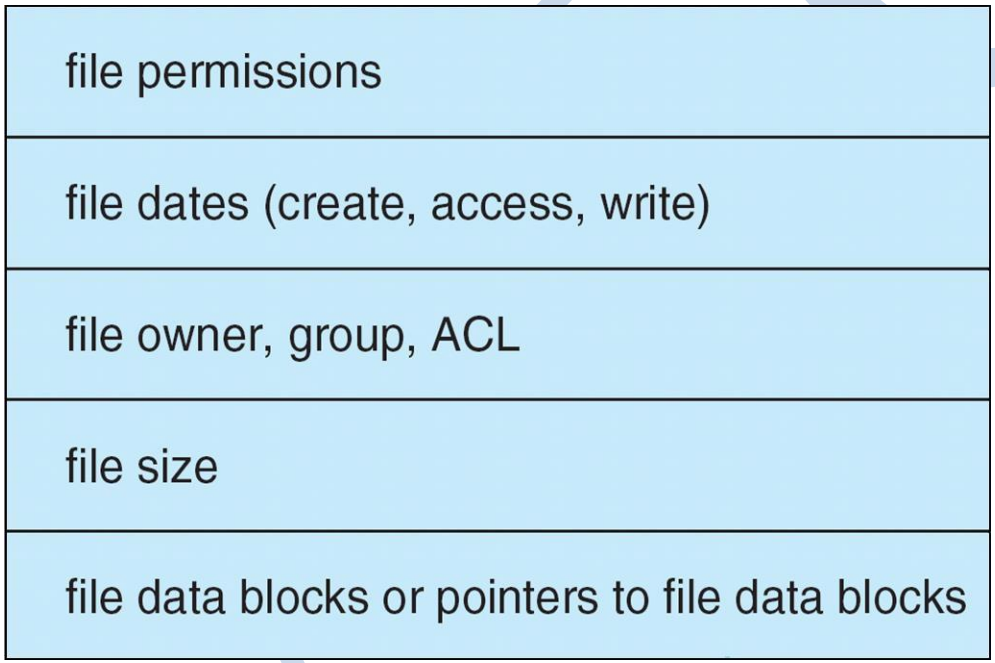
Basic File System - issue command to appropriate device driver

↓ (drive 1, cylinder 44, surface 7, sector 13) to appropriate device driver

I/O Control - device drivers

↓ writes I/O controller's registers to perform appropriate operation

LAYERED FILE SYSTEM CONTD... F LE CONTROL BLOCK



- A file control block (FCB) contains information about the file, including ownership, permissions, and location of the file contents

APEC

LAYERED FILE SYSTEM (CONTINUED)

- Advantage of a layered structure: Duplication of code is minimized
  - Each file system can have its own logical file system and file-organization modules

- Many kinds of file systems are in use today
  - CD-ROMs are written in the ISO 9660 format
  - UNIX uses the UNIX file system (UFS)
  - Windows NT, 2000, and XP support NTFS (New Technology File System)
  - Linux supports over 40 different file systems

APEC

FILE ATTRIBUTES

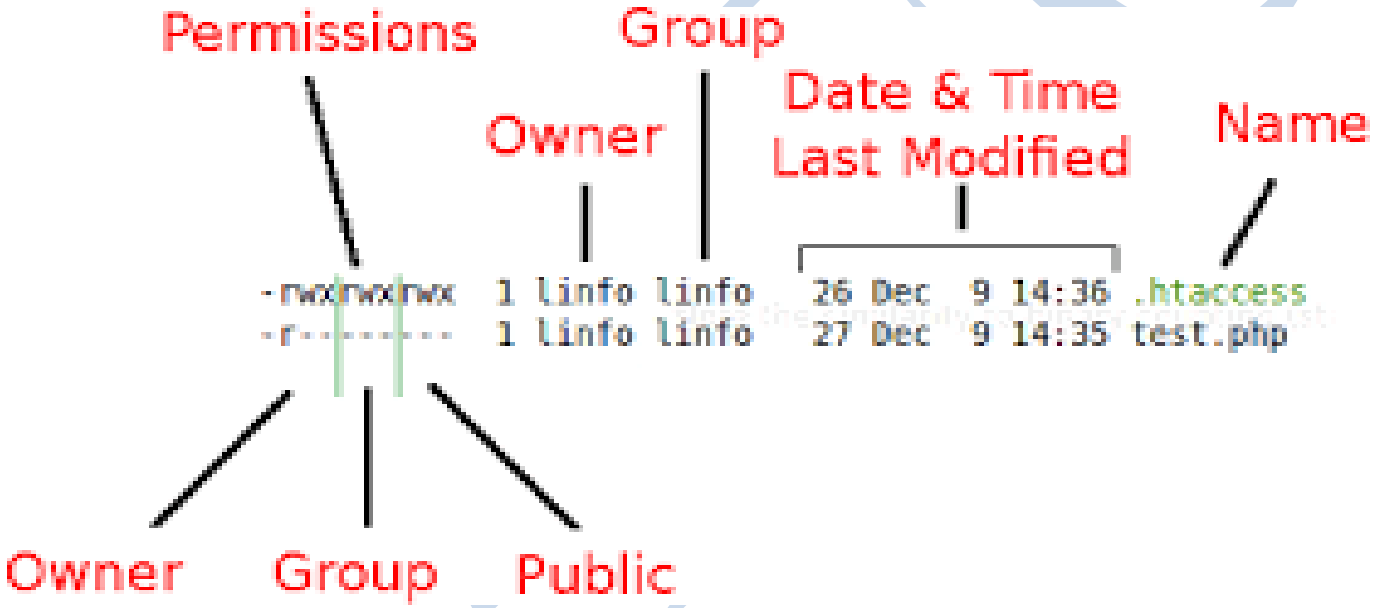
- Name – File name is the only information kept in human-readable form
- Identifier – unique tag (number) identifies file within file system
- Type – needed for systems that support different types
- Location – the information is a pointer to a device, and to the location on device
- Size – current file size (bytes, blocks & words)
- Protection – Access control information, such as who can do reading, writing, executing.
- Time, date, and user identification – The information is kept for creation, last modification, last use etc. These

APEC

information's are kept for protection, security, and monitoring.

APEC

FILE ATTRIBUTES



FILE SYSTEM OPERATIONS

- Create a file
  - Identify the space & create a file.
- Write to a file
  - We need to know the name of the file & what information to be written.
- Read from a file
  - Keep the read pointer to the location, where we want to read.
- Repositioning within the file
  - Current file position pointer is repositioned
- Delete a file
  - Search the directory for the named file and delete it. After deleting, free space are available.
- Truncate a file

APEC

- Truncate a file
  - When the user wants to erase the contents of the file, but keeping its attributes.

APEC

FILE TYPES



file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com-pressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

ACCESS METHODS

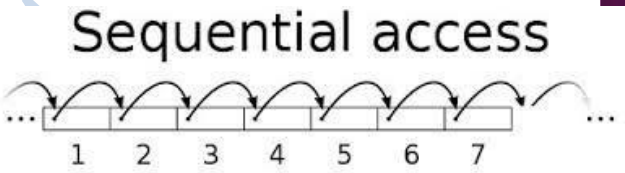
AFEC

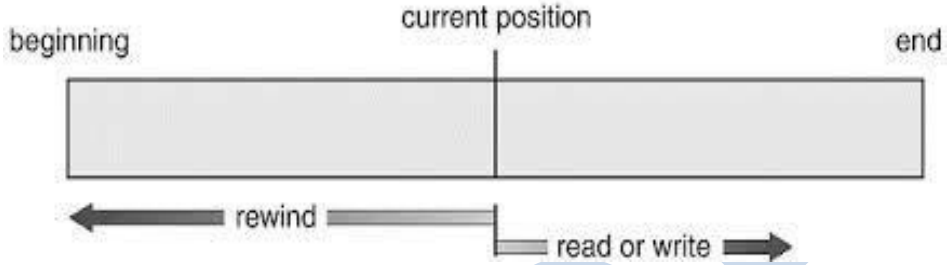
ACCESS METHODS

- Sequential Access
- Direct Access
- Indexed Access

APEC

SEQUENTIAL ACCESS





Definition:

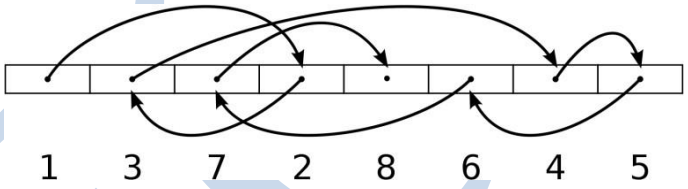
- Information in the file is processed in order, one record after the another.
- Example: Editors & Compilers.

- **Sequential Access operations**
  - ***read next :*** Reads the next portion of the file and automatically advances the File Pointer (FP)
  - ***write next :*** writes data to the file and advances File Pointer (FP)
  - ***Reset***
  - ***skip forward***
  - ***skip backward***

- Initially File Pointer is at the beginning and the program may be able to skip forward (or) backward ‘n’ records.

APEC

DIRECT ACCESS OR RELATIVE ACCESS



- It allows programs to read & write records rapidly in no particular order.

- For direct access(position to n)
  - File is viewed as numbered sequence of blocks or records
  - We may read block 14, then block 53 and then block 97 etc.

APEC

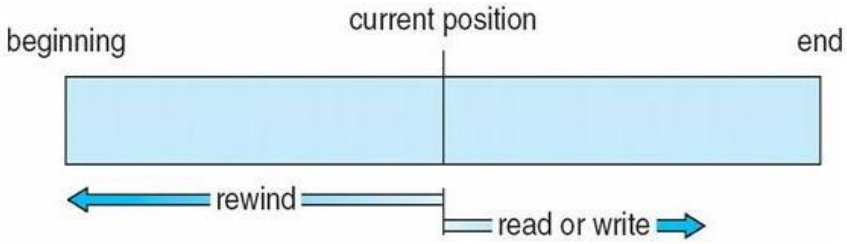
- Direct access files are of great use of immediate access to large amounts of information

APEC

SEQUENTIAL VS DIRECT ACCESS

- **Sequential Access**

  read next
  write next
  reset



- **Direct Access**

  read *n*
  write *n*
  position to *n*
  read next
  write next
  rewrite *n*

  ***n* = relative block number**

The block number provided by the user to the operating system is called relative block number. It acts as a index to the file.

APEC

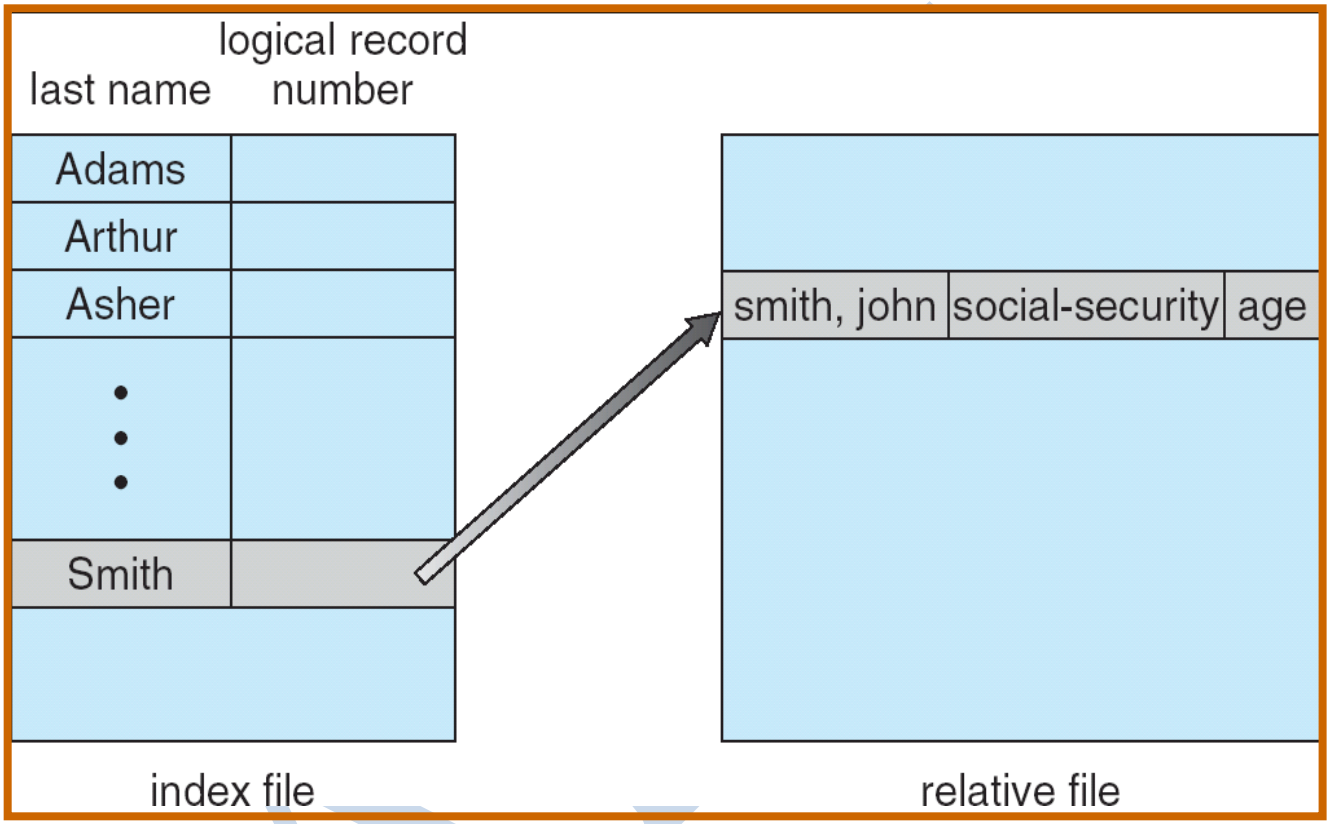
SEQUENTIAL VS DIRECT ACCESS

sequential access	implementation for direct access
reset	*cp* = 0;
read next	*read cp*; *cp* = *cp* + 1;
write next	*write cp*; *cp* = *cp* + 1;

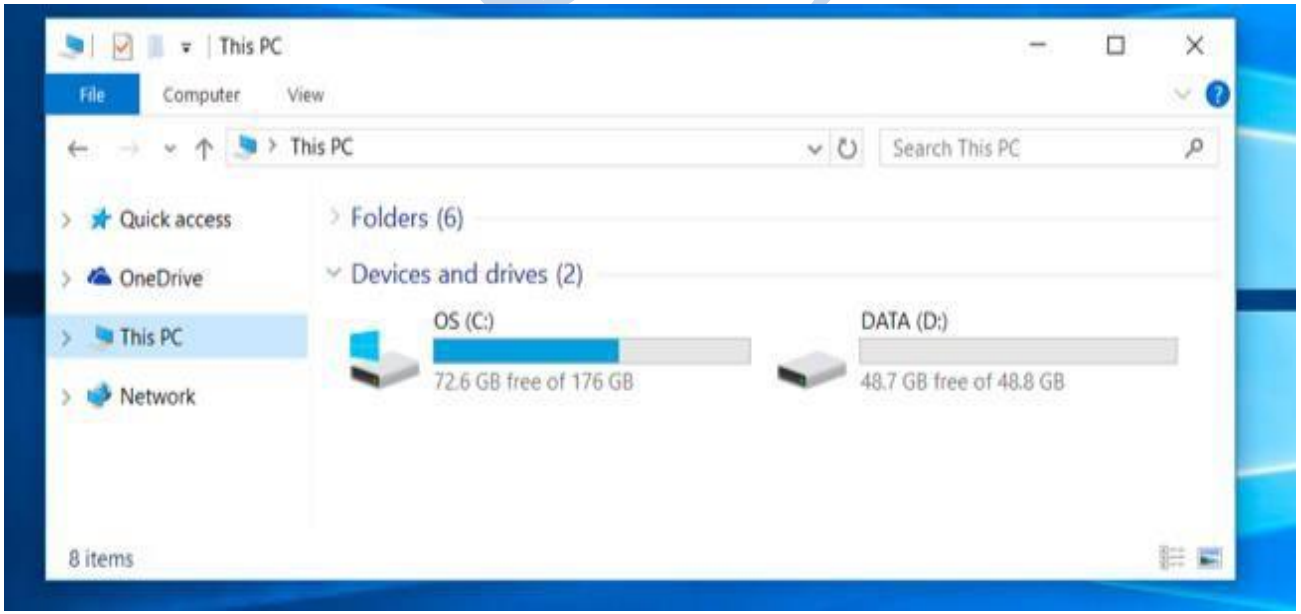
INDEXED ACCESS

- It is done with the help of the index, where the index is like the book index.

APEC


logical record
last name
number
Adams
Arthur
Asher
Smith
smith, john
social-security
age
index file
relative file

DIRECTORY STRUCTURE



DIRECTORY STRUCTURE

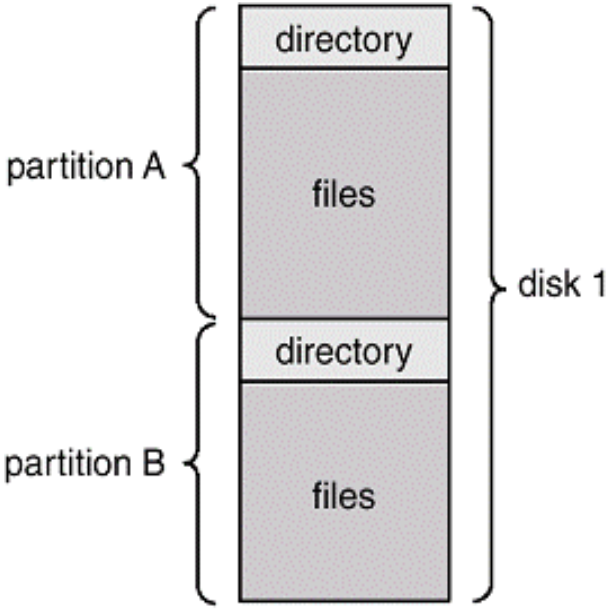


Fig. Single Disk Multiple Partitions

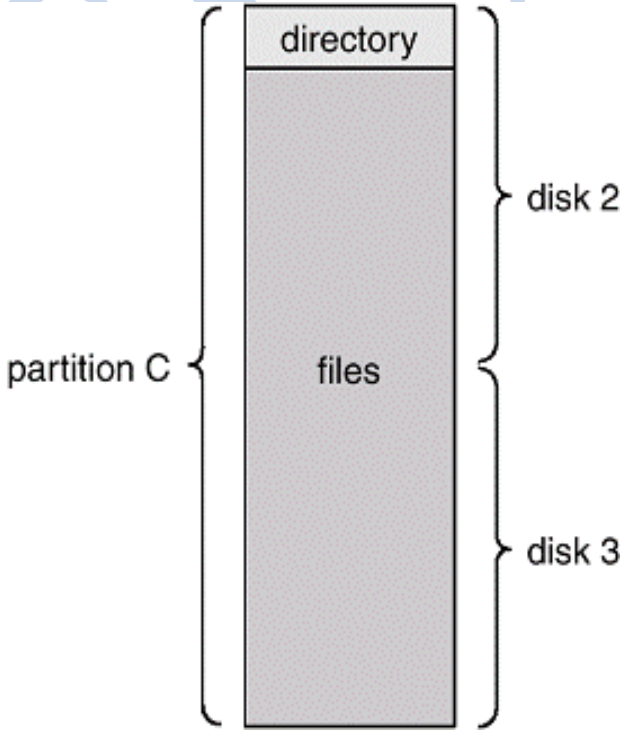


Fig. Multiple Disk Single Partition

Fig. Various File system organization schemes

CONTD...

- To manage enormous amount of data, the hard disk is divided into partitions (or) volumes. Each partition stores information about the files stored in it.
- The information is stored in the directory (or) device directory.

- Different operations in the directory are:
  - Create a file
  - Search a file
  - List a directory
  - Rename a file

APEC

- Delete a file
- Traverse the file system

APEC

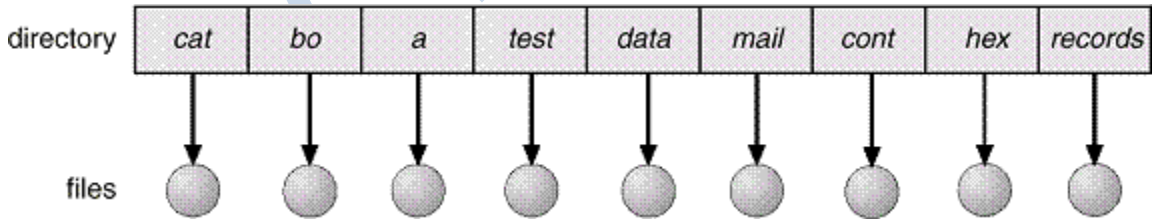
FOUR POSSIBLE APPROACHES

- Single-level Directory
- Two-level Directory
- Tree-structured Directories
- Acyclic Graph Directories

APEC

SINGLE-LEVEL DIRECTORY

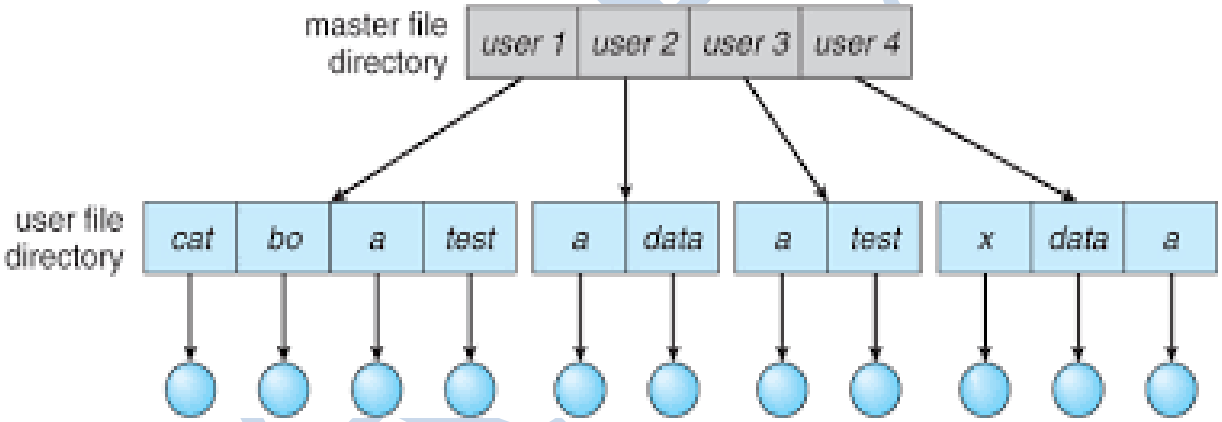
- A single directory maintained for all users
- Advantages
  - Efficiency
- Disadvantages
  - Cannot manage many files in a single directory
  - Cannot have same name for both files



APEC

TWO-LEVEL DIRECTORY

- Two directories are created
  - User File directory (UFD)
  - Master File Directory (MFD)



Advantages:

- Same file name in different directories

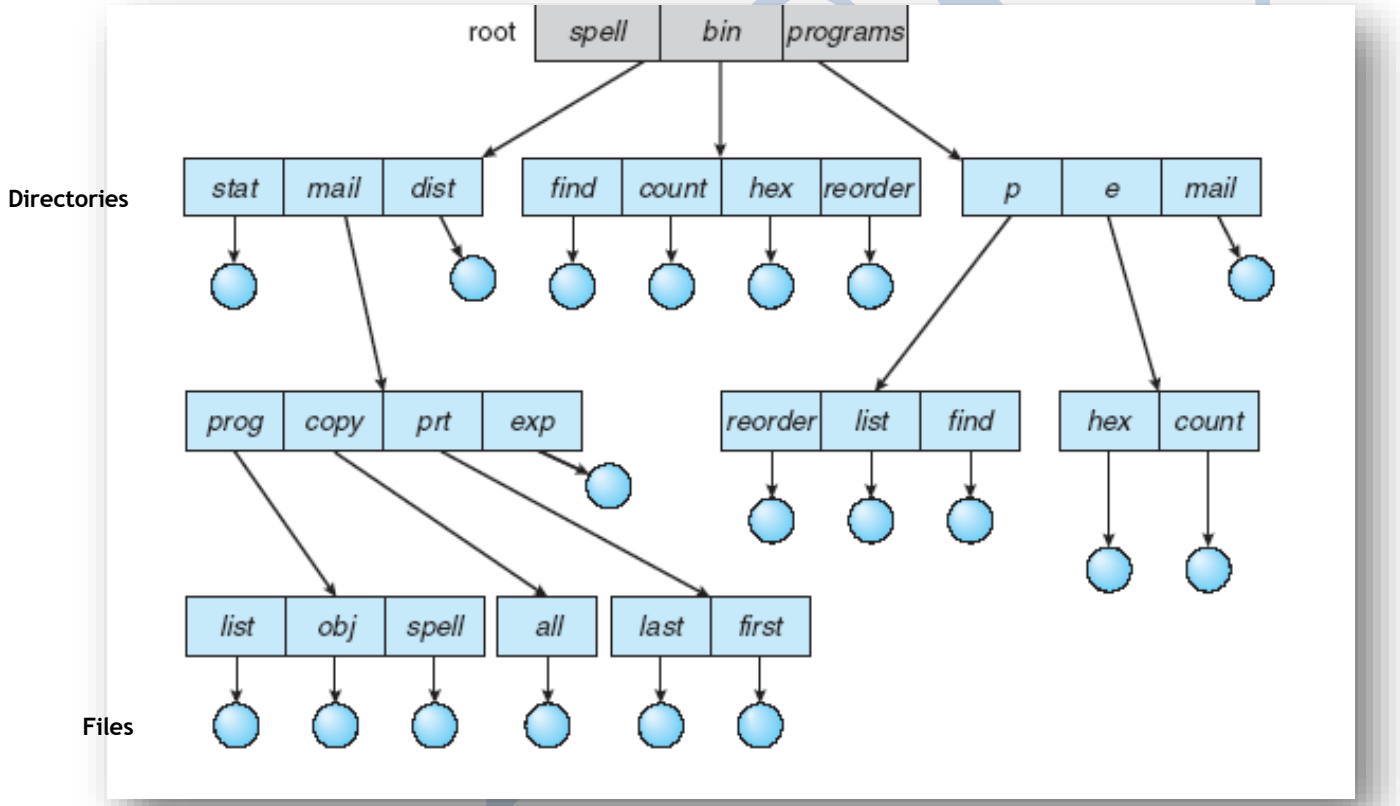
Disadvantages:

- Users in one directory structure is not allowed to access files of another users
- Users must know the path name..
  - **//user2//data//xx.pptx**

APEC

HIERARCHIAL DIRECTORY

- Tree Directory or Tree-Structured directory



TREE-STRUCTURED DIRECTORIES (CONT)

- Advantages
  - Efficient searching – current working directory
  - Naming - same file name in a different directory
  - Grouping capability
- Disadvantages
  - Very much complex

APEC

DEFINITIONS

- **Absolute Path Name:**
  - Begins at the root and follows the path down to the specified file.

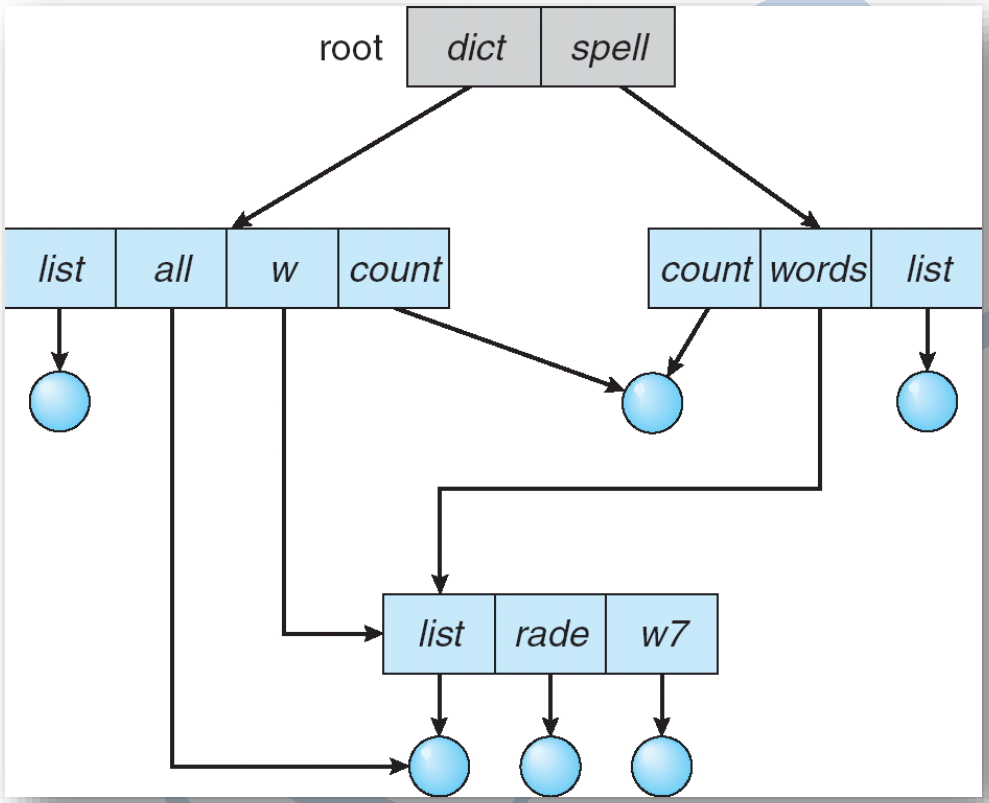
- **Relative Path Name:**
  - Defines the path from the current working directory.

APEC

ACYCLIC-GRAPH DIRECTORIES

- Have shared subdirectories and files
- When 2 programmer's want to work in a joint project. They share common subdirectories.
- Acyclic ➔ graph with no cycles

APEC


root
dict
spell
list
all
w
count
count
words
list
list
rade
w7

ACYCLIC-GRAPH DIRECTORIES (CONT.)

- **Advantages:**
  - Same advantages as tree-structured directory
- **Disadvantage:**
  - Structure is more complex

APEC

FILE SHARING AND PROTECTION

FILE SHARING AND PROTECTION

- Multiple users
- Remote File system
- File Protection

APEC

FILE SHARING

- Sharing of files on multi-user systems is desirable
- Sharing may be done through a **protection scheme**
- On distributed systems, files may be shared across a network
- Network File System (NFS) is a common distributed file-sharing method
- If multi-user system
  - **User IDs** identify users, allowing permissions and protections to be per-user
    Group IDs allow users to be in groups, permitting group access rights

APEC

- Owner of a file / directory
- Group of a file / directory

APEC

FILE SHARING – REMOTE FILE SYSTEMS

- Uses networking to allow file system access between systems
  - Manually via programs like **FTP**
  - Automatically, seamlessly using **distributed file systems**
  - Semi automatically via the **world wide web**

- Networking allows sharing or resource spread within the campus or even around the world. User transfer files between machines like FTP.
- A Distributed File System (DFS) in which remote directories are visible from the local machine.

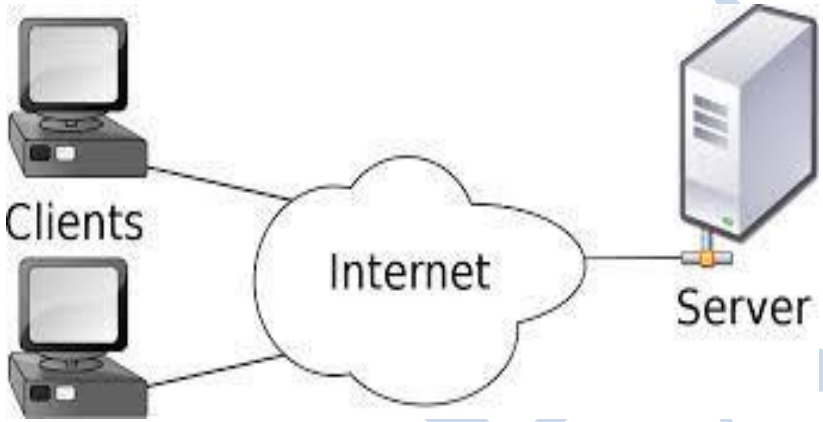
- The information system on the Internet allows documents to be connected to other documents by hypertext links.

APEC

FILE SHARING – REMOTE FILE SYSTEMS

- **Client-server** model allows clients to mount remote file systems from servers
  - Server can serve multiple clients
  - Client and user-on-client identification is insecure or complicated
  - **NFS** is standard UNIX client-server file sharing protocol
  - **CIFS** is standard Windows protocol
  - Standard operating system file calls are translated into remote calls

APEC



APEC

FILE SHARING – REMOTE FILE SYSTEMS

- Distributed Information Systems **(distributed naming services)** such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote

APEC

computing



- Lightweight Directory Access Protocol (LDAP)
- Domain Name System (DNS)
- Network Information Service (NIS)

FILE SHARING – FAILURE MODES

- Remote file systems add new failure modes, due to network failure, server failure
- Recovery from failure can involve state information about status of each remote request
- Stateless protocols such as NFS include all information in each request, allowing easy recovery but less security

APEC

FILE PROTECTION

- File owner/creator should be able to control:
  - what can be done
  - by whom
- Types of access
  - **Read**
  - **Write**
  - **Execute**
  - **Append**
  - **Delete**
  - **List**

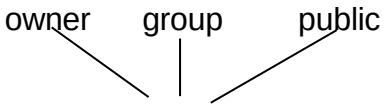
APEC

ACCESS LISTS AND GROUPS

- Mode of access: read, write, execute
- Three classes of users

			RWX
a) **owner access**	7	⇒	1 1 1
b) **group access**	6	⇒	RWX 1 1 0
c) **public access**	1	⇒	RWX 0 0 1

- Ask manager to create a group (unique name), say G, and add some users to the group.
- For a particular file (say *game*) or subdirectory, define an appropriate access.



chmod 761 game

Attach a group to a file

chgrp G game

APEC

FILE SYSTEM MOUNTING

APEC

MOUNTING AND UNMOUNTING FILE SYSTEMS

- 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.

APEC

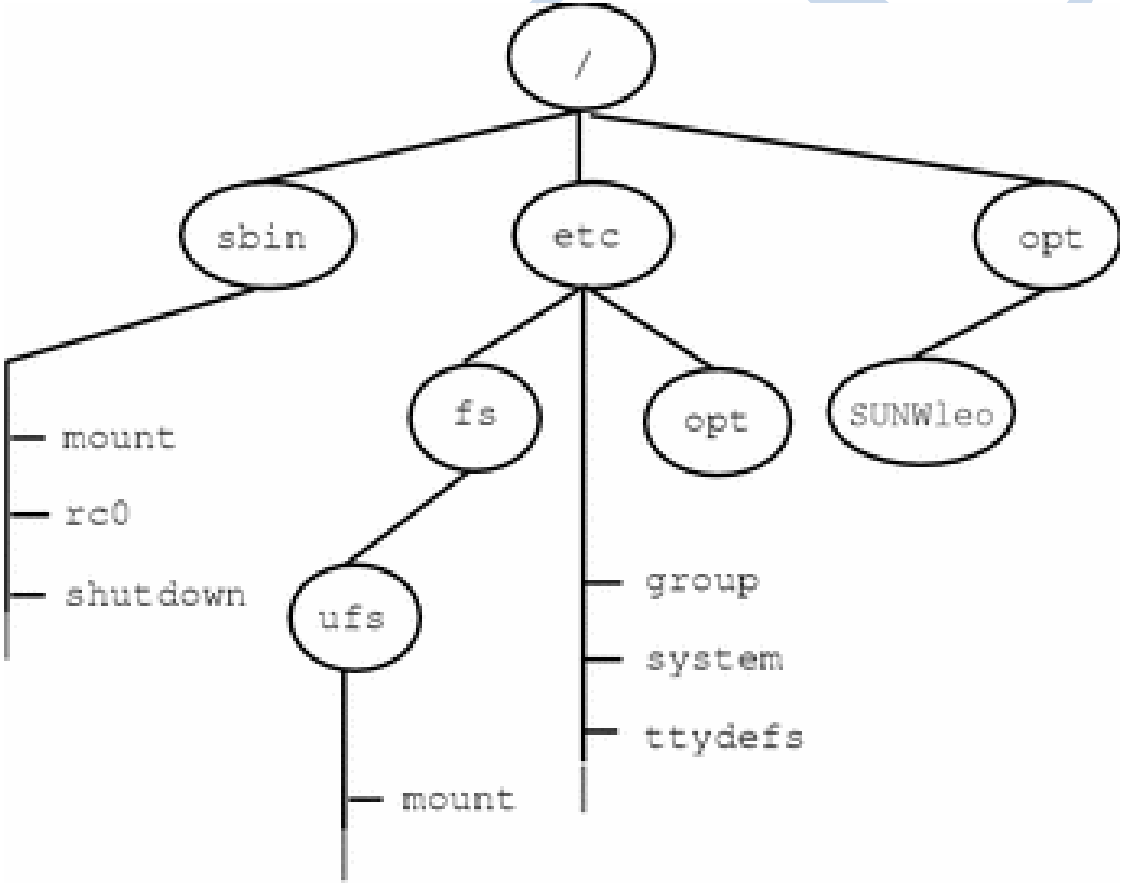
CONTD...

- Mounting is the attaching of an additional file system to the currently accessible file system of a computer.

- Mounts may be local or remote.
  - A **local mount** connects disc drives on one machine
  - A **remote mount** uses **Network File System (NFS)** to connect to directories on other machines.

APEC

LOCAL FILE SYSTEM, STARTING WITH A ROOT (/) FILE SYSTEM AND SUBDIRECTORIES SBIN, ETC, AND OPT.



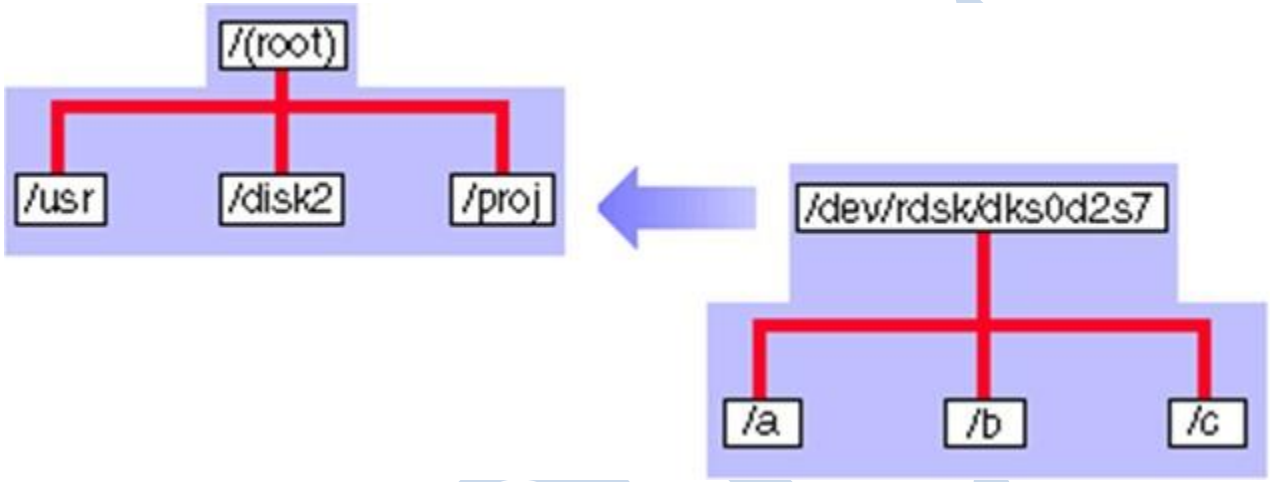


Fig. Before Mounting

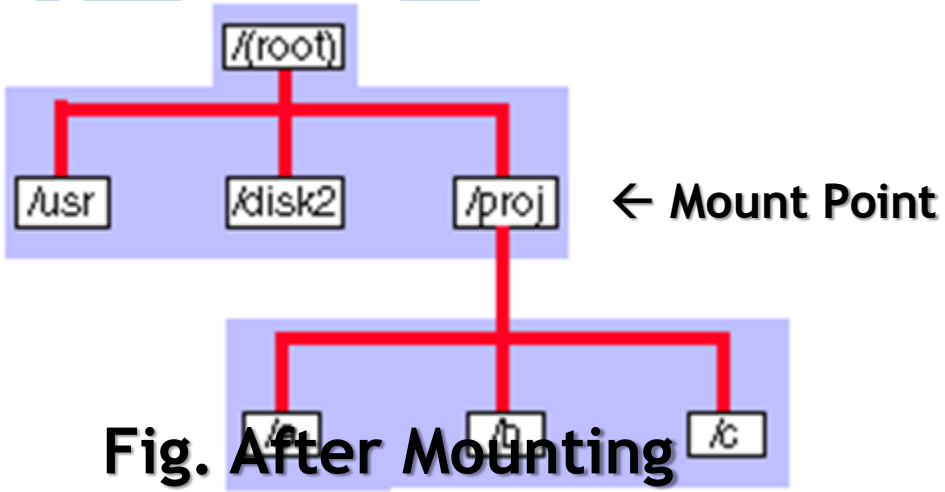


Fig. After Mounting

APEC

EXPLANATION

- When a file system is mounted, the name of the device file for the file system (/dev/rdsk/dks0d2s7) and the name of a directory (/proj) are given.
- This directory, /proj, is called a mount point and forms the connection between the file system containing the mount point and the file system to be mounted.
- Hence the files a, b, and c in the /dev/rdsk/dks0d2s7 file system become /proj/a, /proj/b, and /proj/c.

AREC

HOW TO DO?

Mounting

- Manually with the mount command.
- Automatically when the system is booted, using information in the file /etc/fstab.
- Automatically when the file system is accessed (called automounting, which applies to NFS (remote) file systems only.

Unmounting

- Shut down the system (file systems are unmounted automatically)

AREC

- Manually unmount file systems with the umount command.

APEC

FILE SYSTEM IMPLEMENTATION

AREC

OVERVIEW

- Several on-disk and in-memory structures are used to implement the file system.

1

- **In disk Information:**
  - The file system contain information about how to boot the operating system, the total number of blocks, the total number of free blocks etc.
    - Boot control Block
    - Volume control Block

APEC

BOOT CONTROL BLOCK & VOLUME CONTROL BLOCK

- **Boot control Block**
  - Contain information needed by the system to boot the operating system
  - It's the typically the 1st block.
  - In UFS, its called boot block
  - In NTFS, its partition boot sector
- **Volume control Block**
  - Contains partition details, such as number of blocks, free block counts, free block pointers
  - In UFS, its called super block and inode.

APEC

- In NTFS, its stored in master file table

Note:

- UFS ➔ Unix File System
- NTFS ➔New Technology File System is a proprietary file system developed by Microsoft

APEC

ON-DISK STRUCTURES CONTD…

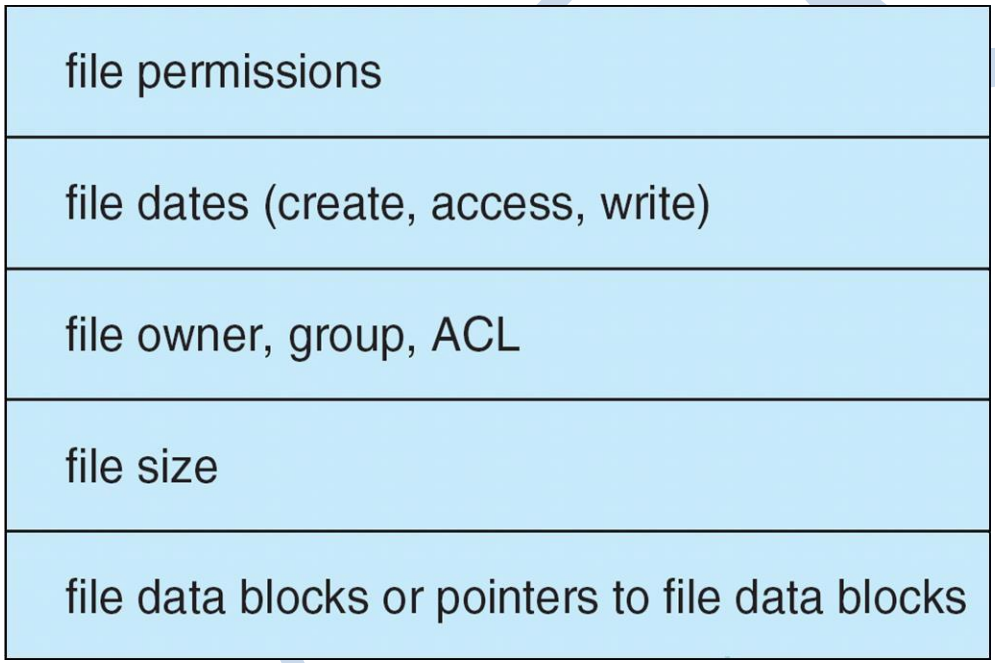
- A directory structure per file system is used to organize the files
  - In UFS, this includes the file names and associated inode numbers
  - In NTFS, it is stored in the master file table
- A File Control Block (FCB) contains many details about the file
- FCB is individual for each file

UFS
NTFS

- In UFS, this is called the inode
- In NTFS, this information is stored in the master file table

APEC

FILE CONTROL BLOCK



- A file control block (FCB) contains information about the file, including ownership, permissions, and location of the file contents
- Contains information about inode number, permissions, size, dates

APEC

2

- **In Memory Information** :
  - Contains information for file system management & performance improvement via caching.
  - The data are loaded at mount time, updated during file system implementations & discarded at dismount.
    - An in-memory mount table.
    - An in-memory directory cache of recently accessed directory information.
    - A system-wide open file table, containing a currently opened FCB.
    - A per-process open file table, containing a pointer to the system open file table as well as some other information.

APEC

IN-MEMORY FILE SYSTEM STRUCTURES

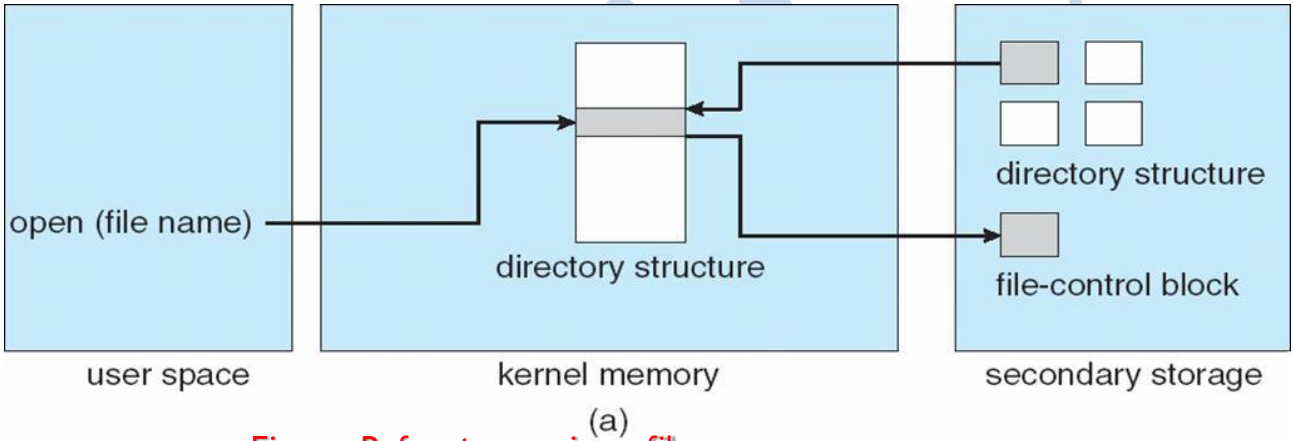


Figure: Refers to opening a file

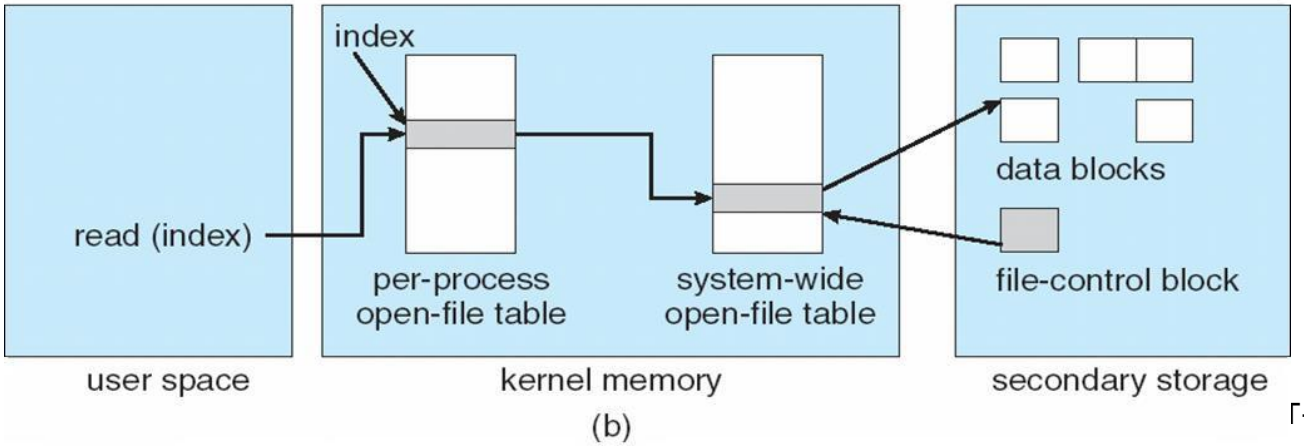


Figure: Refers to reading a file

APEC

VIRTUAL FILE SYSTEMS

APEC

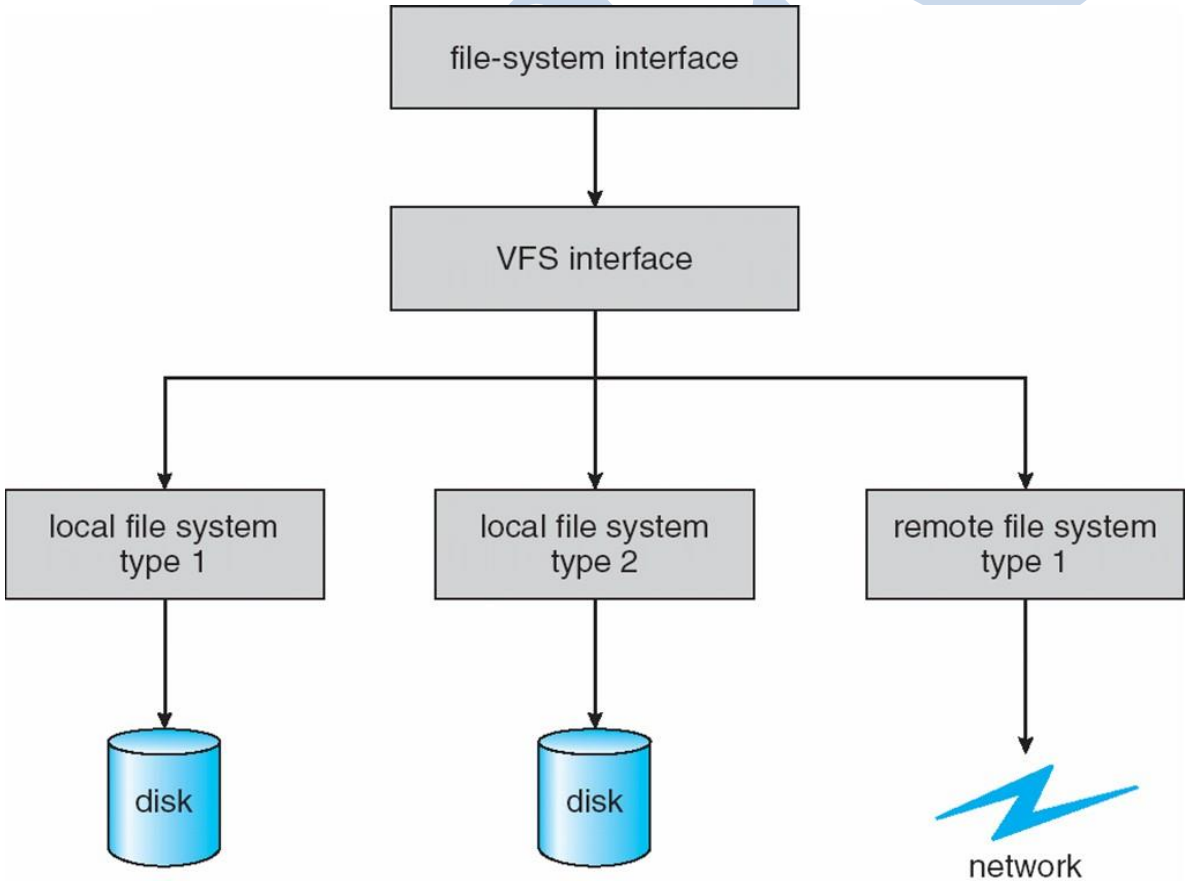
WHY VFS?

- To allow operating system for supporting multiple file systems. Most operating system use object oriented technique to simplify, organize & modularize the implementation.

- Users can access the files that are contained within multiple file systems on the local disk or even the file system across the network.

APEC

VIRTUAL FILE SYSTEMS (CONT.)



EXPLANATION

- The file system implementation consists of 3 layers.
- 1st layer – File system interface based on open(), read(), write(), close()
- 2nd layer – Virtual file system layer (VFS) layer.
- 3rd layer – the type of the local file system

APEC

FUNCTIONS OF 2ND LAYER - VFS LAYER

- Separates file-system generic operations from implementation details by defining clear VFS interface.
- Provides a mechanism for uniquely representing a file throughout a network.

APEC

VFS ARCHITECTURE IN LINUX

The 4 main object types defined by Linux VFS are:

- Inode object ➔ represents individual file
- File object ➔ represents open file
- Superblock object ➔ represents entry file system
- Dentry object ➔ represents individual directory entry

Some of the operations are:

- int open(..) ➔ open a file
- ssize_t read(..) ➔ read from a file
- ssize_t write(..)
- int mmap(..)

- ssize_t write(..) ➔ write to a file
- int mmap(..) ➔ memory map a file

APEC

DIRECTORY IMPLEMENTATION

APEC

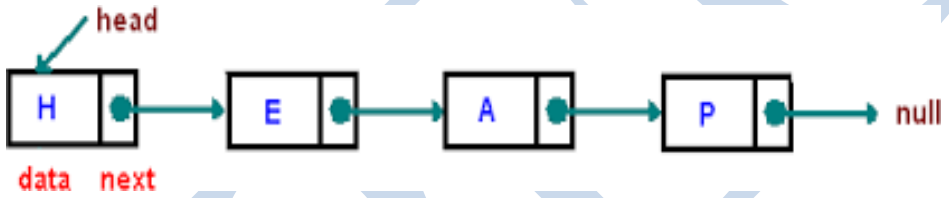
OVERVIEW

- The efficiency, performance and reliability of a file system are directly related to the directory management and directory allocation algorithms.
- Directory management algorithms are
  - Linear list
  - Hash table

APEC

LINEAR LIST

Direct Indexing of a Linear List



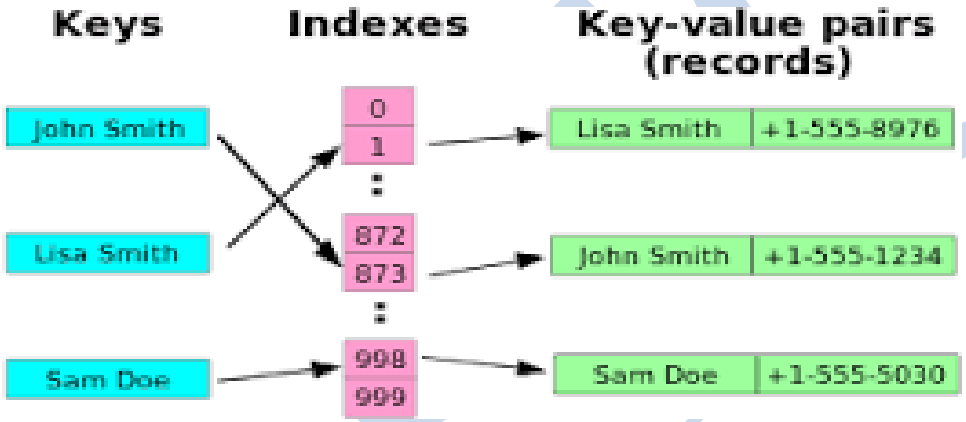
- Use linear list of file names with pointers to data blocks.
- Simple to program
- Time-consuming to search because it is a linear search

- More complicated in creating and deleting files

APEC

HASH TABLE

List Indexing via Hash function



- Takes a value computed from the file name and returns a pointer to the file name in the linear list
- Greatly reduces the directory search time

- Can result in collisions – situations where two file names hash to the same location
- Has a fixed size (i.e., fixed number of entries)

APEC

ALLOCATION METHODS

ALLOCATION METHODS - OVERVIEW

- An important function of the file system is to manage the space on the secondary storage. It is necessary to keep track of free blocks and allocated blocks.

- **Major issues:**
  - Optimum usage of the available disk space
  - Fast accessing of files

- **Types of allocation methods**
  - Contiguous Allocation

- Linked Allocation
- Indexed Allocation

APEC

CONTIGUOUS ALLOCATION

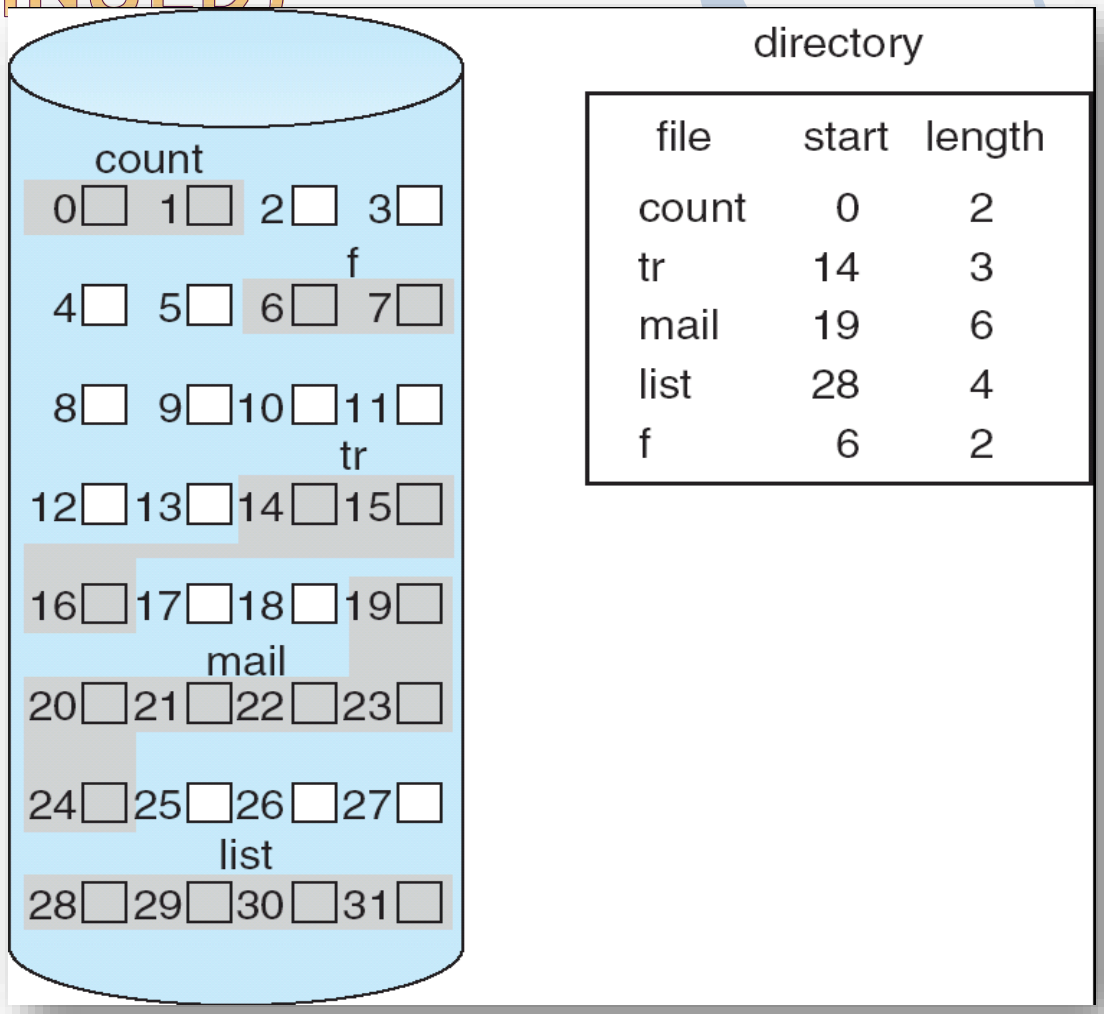
- Contiguous allocation requires that each file, occupy a set of contiguous blocks on the disk.
- Assuming only 1 job is accessing the disk, once the first block “b” is accessed, accessing “b+1” requires no head movement normally.
- Head movement is required only when the head is currently at the last sector of the cylinder and moves to the first sector of the

next cylinder.

- Hence, minimal seek time is achieved.

APEC

CONTIGUOUS ALLOCATION (CONTINUED)



APEC

CONTIGUOUS ALLOCATION -

Advantages & Disadvantages

ADVANTAGES

- Accessing a file is easy – only need the starting location (block #) and length (number of blocks)

DISADVANTAGES

- Finding space for a new file is difficult(first fit, best fit, etc.)
- More External fragmentation (free space is broken into small unusable chunks)
- Hence, compaction is needed.
- Finding space for a file, when the file grows is difficult.

APEC

LINKED ALLOCATION

- Solves the problems of contiguous allocation
- Each file is a linked list of disk blocks: blocks may be scattered anywhere on the disk
- The directory contains a pointer to the first and last blocks of a file
  - For example, a file of five blocks start at block 9, continue to block 16, then block 1, then block 10, and finally block 25.
- Each block contains the pointer to the next block.

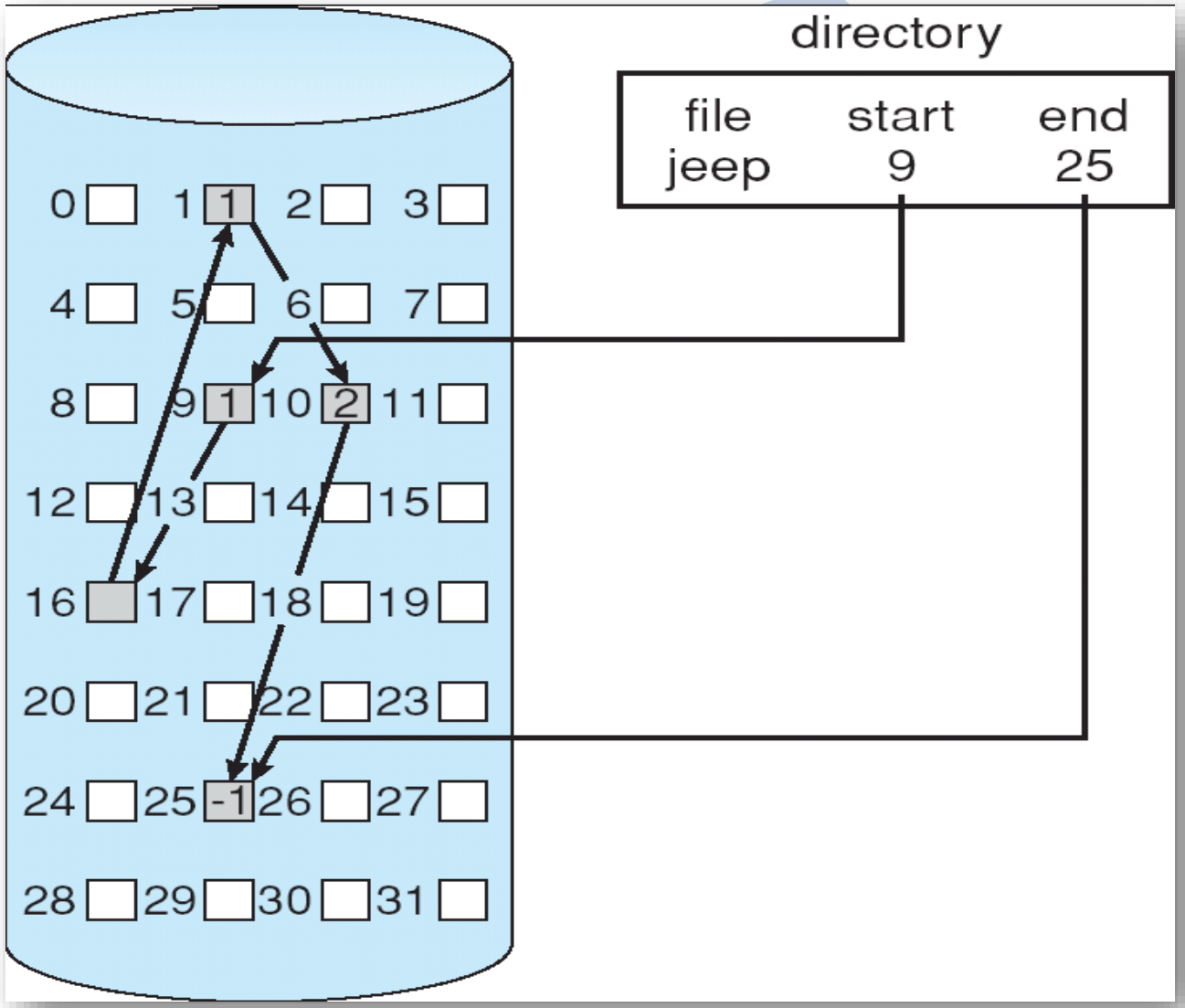
Operations:

- Reading - just moving from block to block
- Writing - find a free block and link to the end of file.
- Creating - create a new entry in the directory

APEC

LINKED ALLOCATION (CONTINUED)

APEC


directory
file
start
end
jeep
9
25
0
1
1
2
3
4
5
6
7
8
9
1
10
2
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
-1
26
27
28
29
30
31

LINKED ALLOCATION (CONTINUED)

Advantages

- No external fragmentation
- The size of a file need not be declared when the file is created
- A file can continue to grow as long as free blocks are available

Disadvantages

- Very useful only for sequential access , but not suitable for direct access
- If any error occurs in the pointer, it may point to

(A partial solution is a doubly linked list).

APEC

the free-space list or to the blocks of another file

APEC

LINKED ALLOCATION (CONTINUED)

- One variation on linked allocation is the **file allocation table (FAT)** used by MS-DOS
  - The FAT has one entry for each disk block and is indexed by block number
  - The directory entry contains the block number of the first block of the file
  - The table entry indexed by the block number contains the block number of the next block in the file
  - The last block contains a special end-of-file value as the table entry
  - Unused blocks are indicated by a zero table value

APEC

- To allocate a new block to a file
  - Find the first zero-valued table entry

APEC

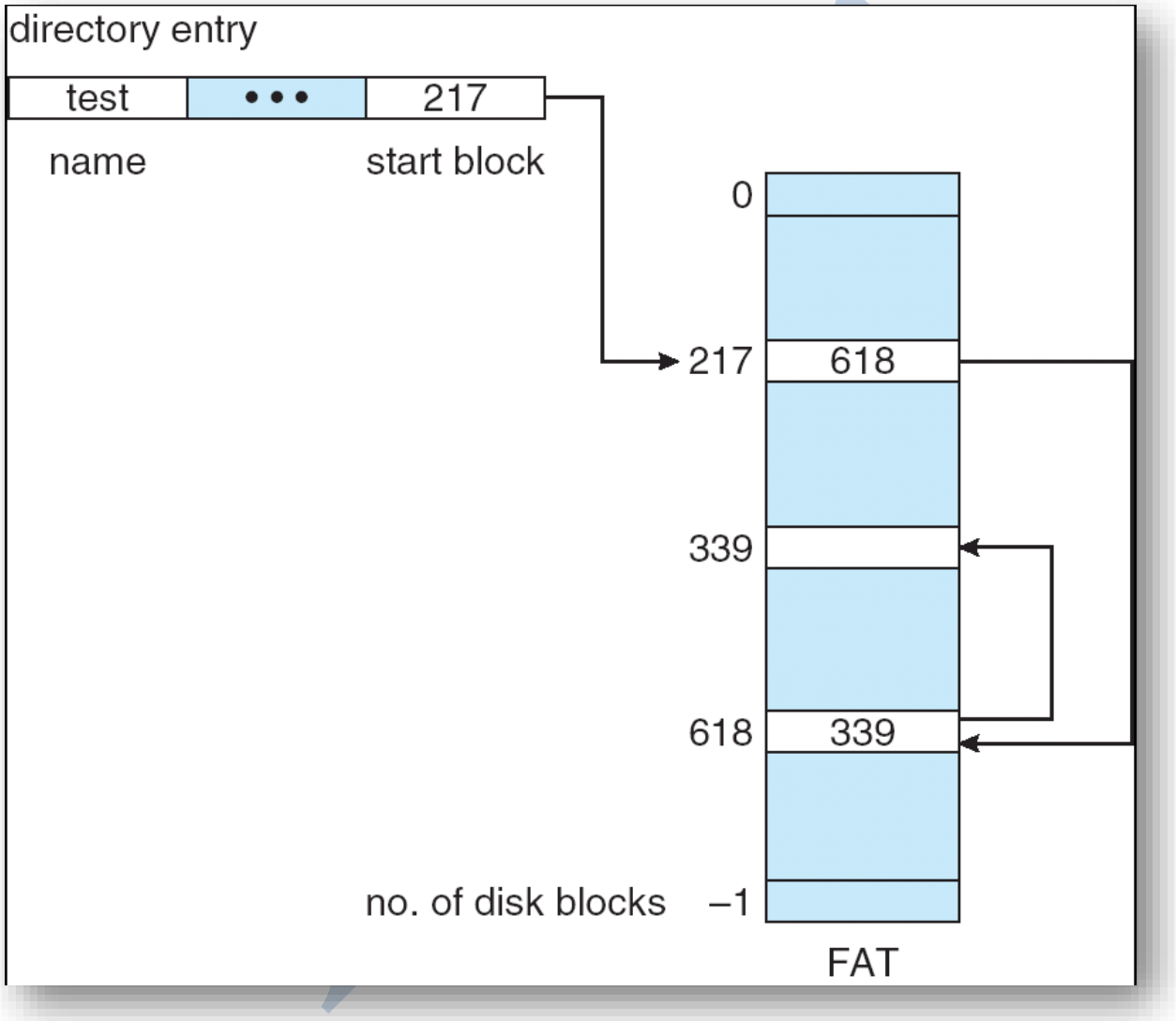
LINKED ALLOCATION (CONTINUED)

- To allocate a new block to a file
  - Find the first zero-valued table entry
  - Replace the previous end-of-file value with the address of the new block
- Disadvantage – can result in a significant number of disk head seeks
- Advantage – random-access time is improved because the FAT can be checked

APEC

F LE-ALLOCATION TABLE (FAT)

APEC


directory entry
test
• • •
217
name
start block
0
217
618
339
618
339
no. of disk blocks
–1
FAT

INDEXED ALLOCATION

- It supports both sequential & indexed access.
- In indexed allocation, the blocks of the file are scattered all over the disk in the same manner as the linked allocation. However, the pointers to the blocks are brought together at one location called Index Block.

- Each file has its own index block, which is an array of disk-block addresses
- Each entry in the index block points to the corresponding block of the file
- The directory contains the address of the index block

APEC

- Finding and reading a specific block in a file only requires the use of the pointer in the index block

APEC

INDEXED ALLOCATION (CONTINUED)

Advantages

- Supports both direct access & direct access without suffering from external fragmentation

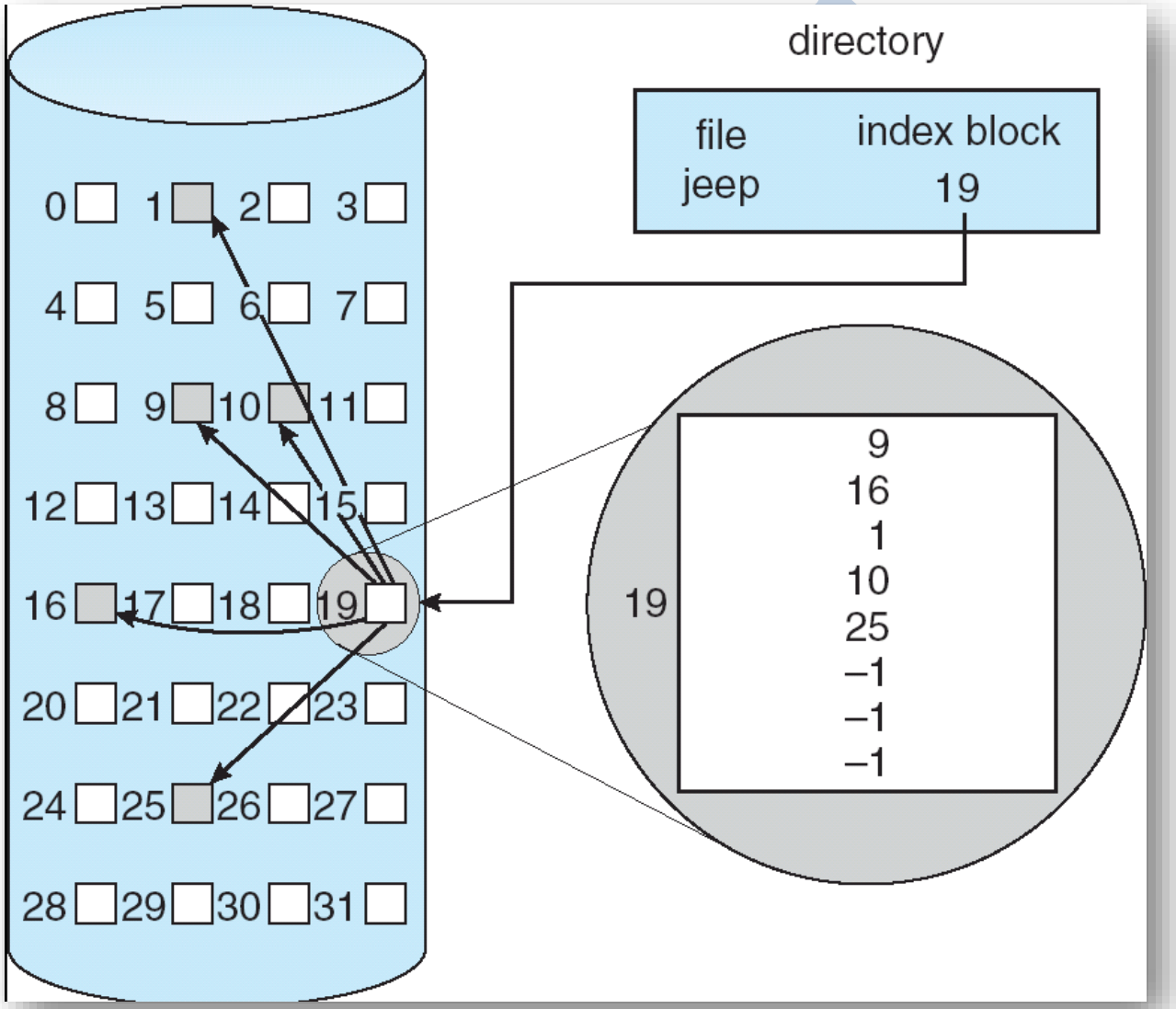
Disadvantages

- Need additional space for index block

APEC

EXAMPLE OF INDEXED ALLOCATION

APEC


directory
file
jeep
index block
19
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
19
9
16
1
10
25
–1
–1
–1

SIZE OF THE INDEX BLOCK

What is the Size of the index block?

- Approach #1: Linked scheme
  - Several index blocks can be linked together
- Approach #2: Multilevel index scheme
  - A first-level index block points to a set of second-level index blocks
- Approach #3: Combined scheme
  - Used in the UNIX file system (UFS)
  - A specific set of pointers points directly to file blocks

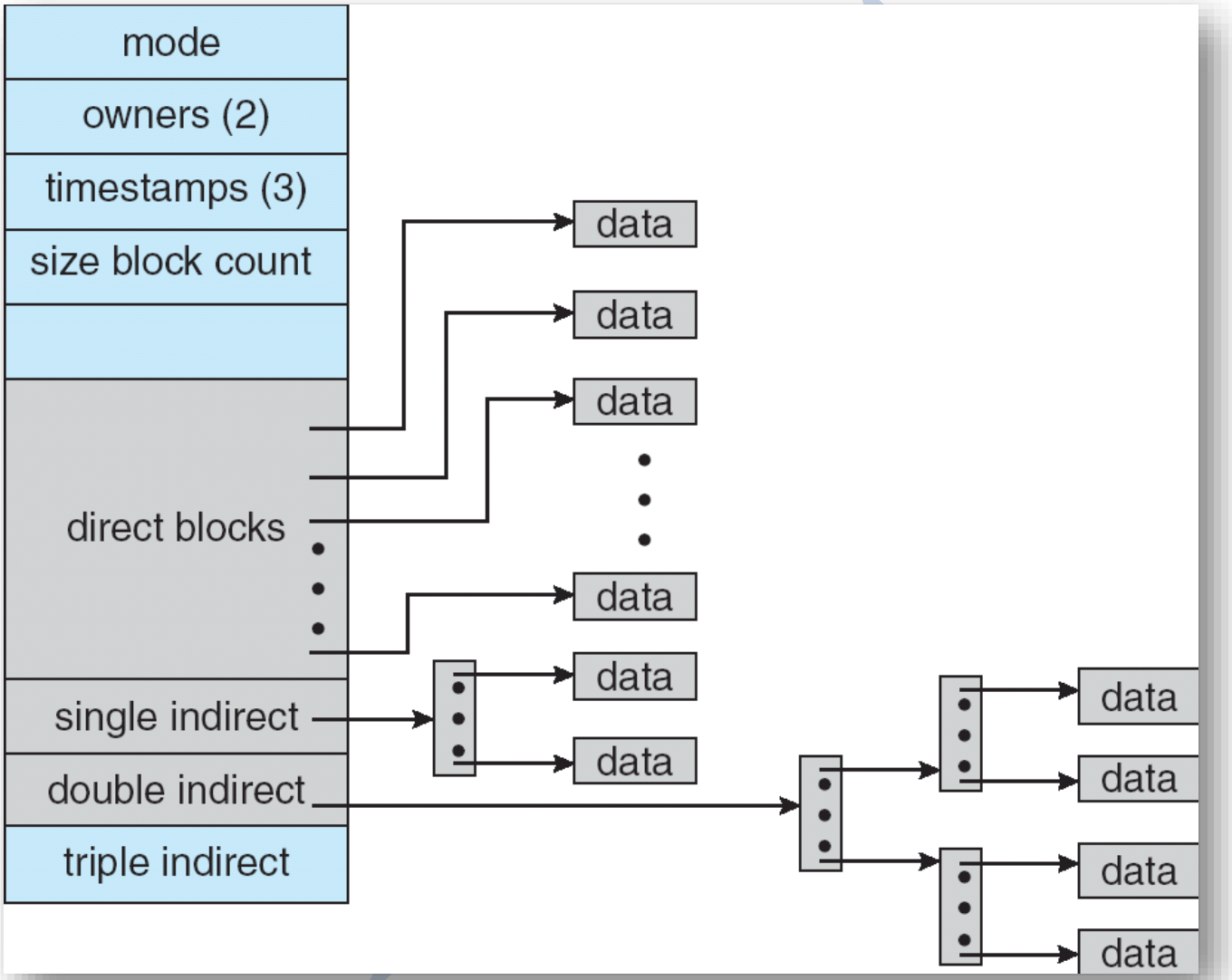
APEC

- Special pointers point to a single indirect block, a double indirect block, and a triple indirect block
- It is used when File size in terabytes

APEC

COMBINED SCHEME: UNIX (4K BYTES PER BLOCK)

APEC


mode
owners (2)
timestamps (3)
size block count
direct blocks
single indirect
double indirect
triple indirect
data
data
data
data
data
data
data
data
data
data

FREE-SPACE MANAGEMENT

APEC

FREE SPACE MANAGEMENT

- Whenever a new file is created, it is allotted some space from the available free space on the disk. The free space may be free space or left over space by the deleted files.
- Hence, file system maintains a **Free Space List.**

- **Various methods** to implement free space are
  - Bit Vector
  - Linked List
  - Grouping

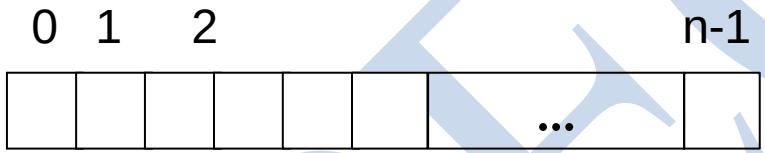
APEC

- Counting

APEC

BIT VECTOR APPROACH

- Free-space blocks are tracked via a bit vector (where *n* = #blocks)



$$\text{bit}[i] = \begin{cases} 0 \Rightarrow \text{block}[i] \text{ allocated} \\ 1 \Rightarrow \text{block}[i] \text{ free} \end{cases}$$

- For example, consider a disk where blocks 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 17, 18,25,26, and 27 are free and the rest of the blocks are allocated.
- The free-space bit map would be 001111001111110001100000011100000 ...

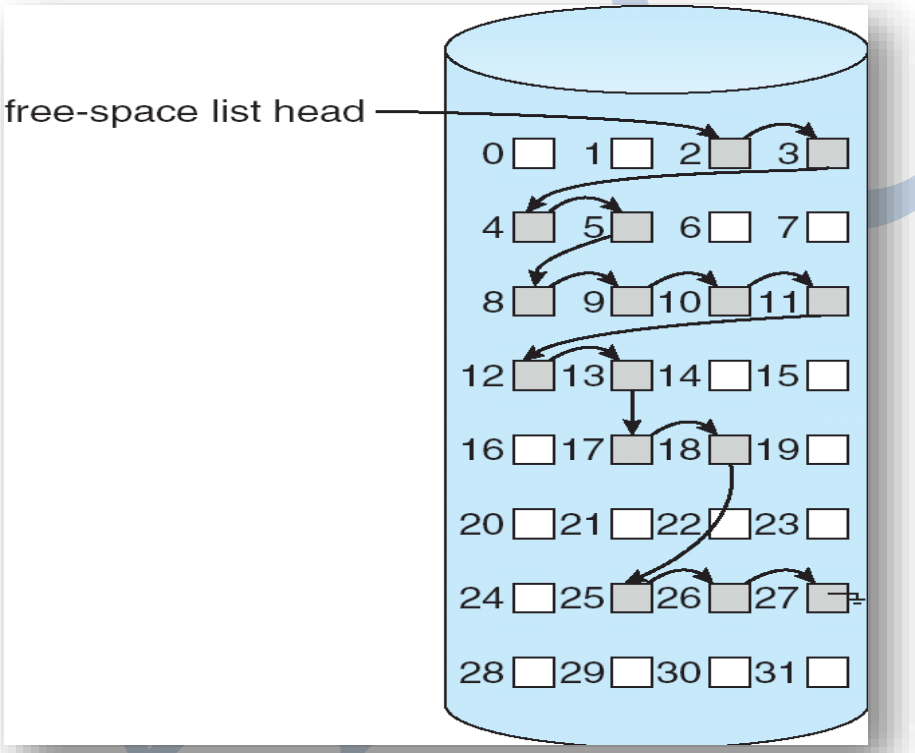
Calculating the number of a free block:

(number of 0-value words) * (number of bits per word) + offset of first 1 bit in a word

- Advantages: Simplicity
- Disadvantages: Need to keep the entire vector in main memory

APEC

LINKED LIST APPROACH



- Link together all the free disk blocks and keep a pointer to the first free block

- Advantage: no waste of space
- Disadvantage: cannot get contiguous blocks easily for a file

APEC

GROUPING

- Keep a linked list of index blocks.
- Each index block contains:
  - Addresses of the free blocks
  - A pointer to the next index block

  - Store the addresses of n free blocks in the first free block
  - The first n – 1 of these blocks are actually free
  - The last block contains the addresses of another n free blocks

- Advantage: Free space can be found easily

APEC

- Disadvantage: Extra Space for index block

APEC

COUNTING

- Instead of keeping a list of addresses of n free blocks, it is more convenient to keep the address of the first free blocks and the "n" number of free contiguous blocks that follow the first block.

APEC

UNIT IV COMPLETED

AEC