OPERATING SYSTEM
(4330703)
CHAPTER-4
File Management System

A.K.PANCHASARA

COMPUTER ENGINEERING DEPARTMENT

A.V.P.T.I RAJKOT

Content

- Disk Organization
  - ✓ Physical Structure
  - ✓ Logical Structure
  - ✓ Addressing
- File Management
  - ✓ User view of File system
  - ✓ Attributes and Operation
  - ✓ File system Design
  - ✓ Disk Space
- Directory Structure
- Security and Protection

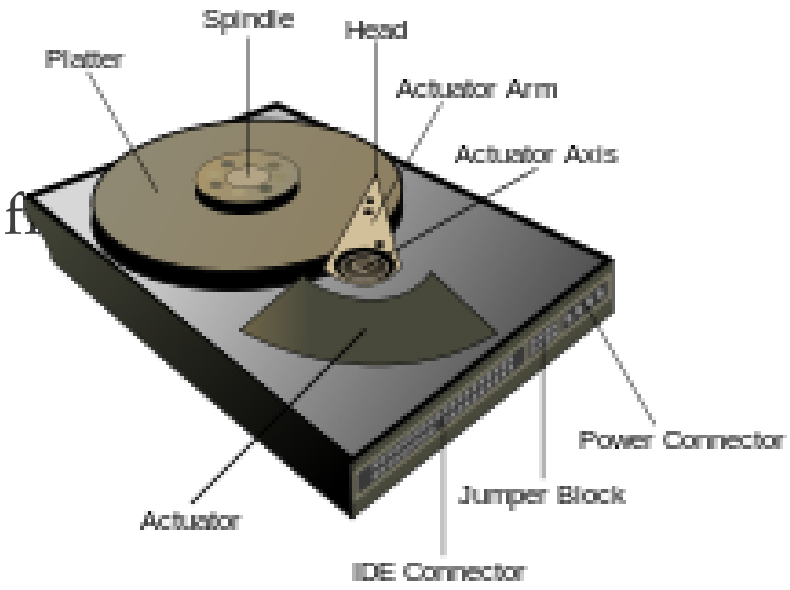
Disk Organization

This section covers different aspects of Disk…

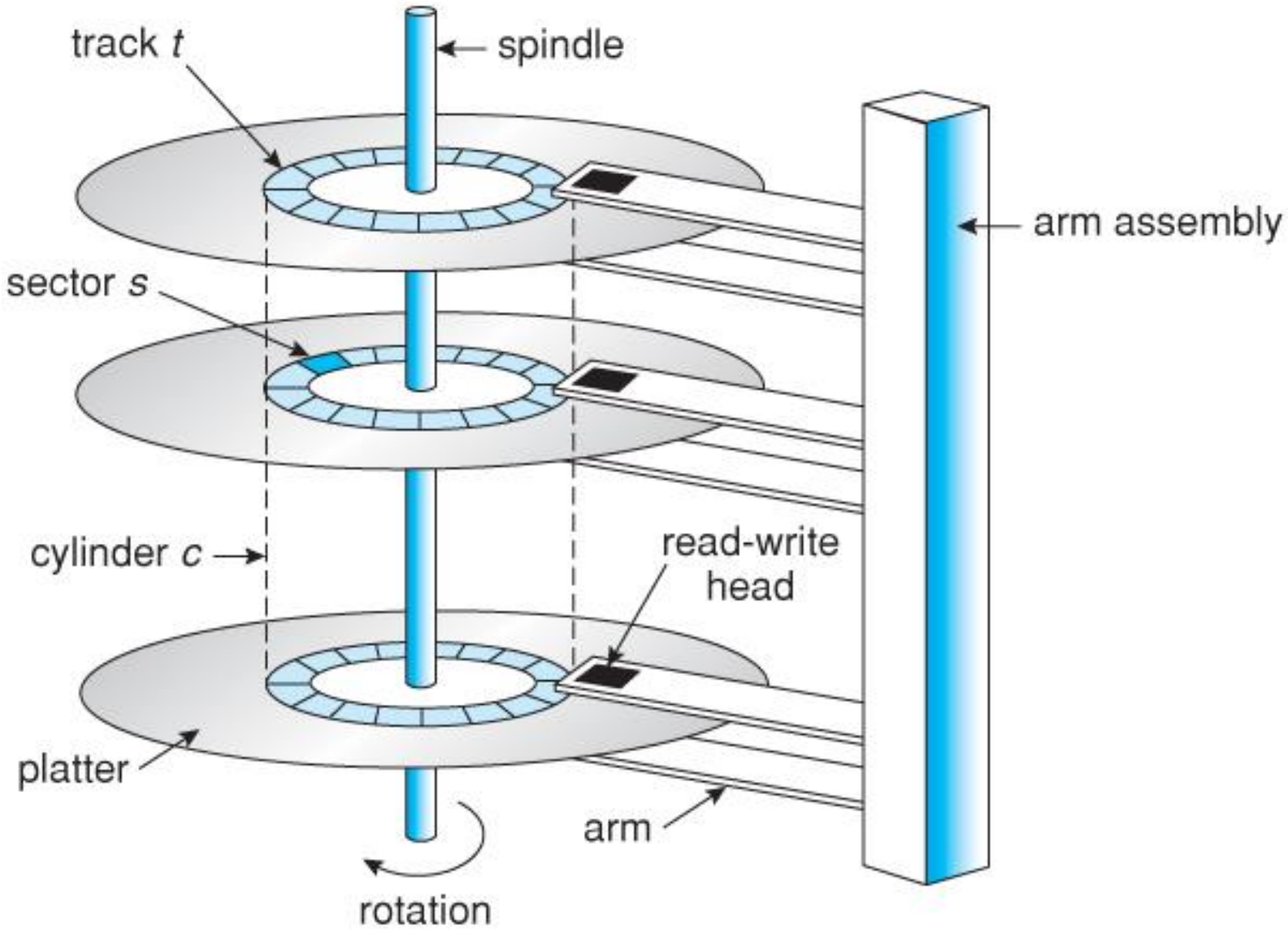
1. Physical Organization
2. Logical Organization
3. Addressing

Physical Structure

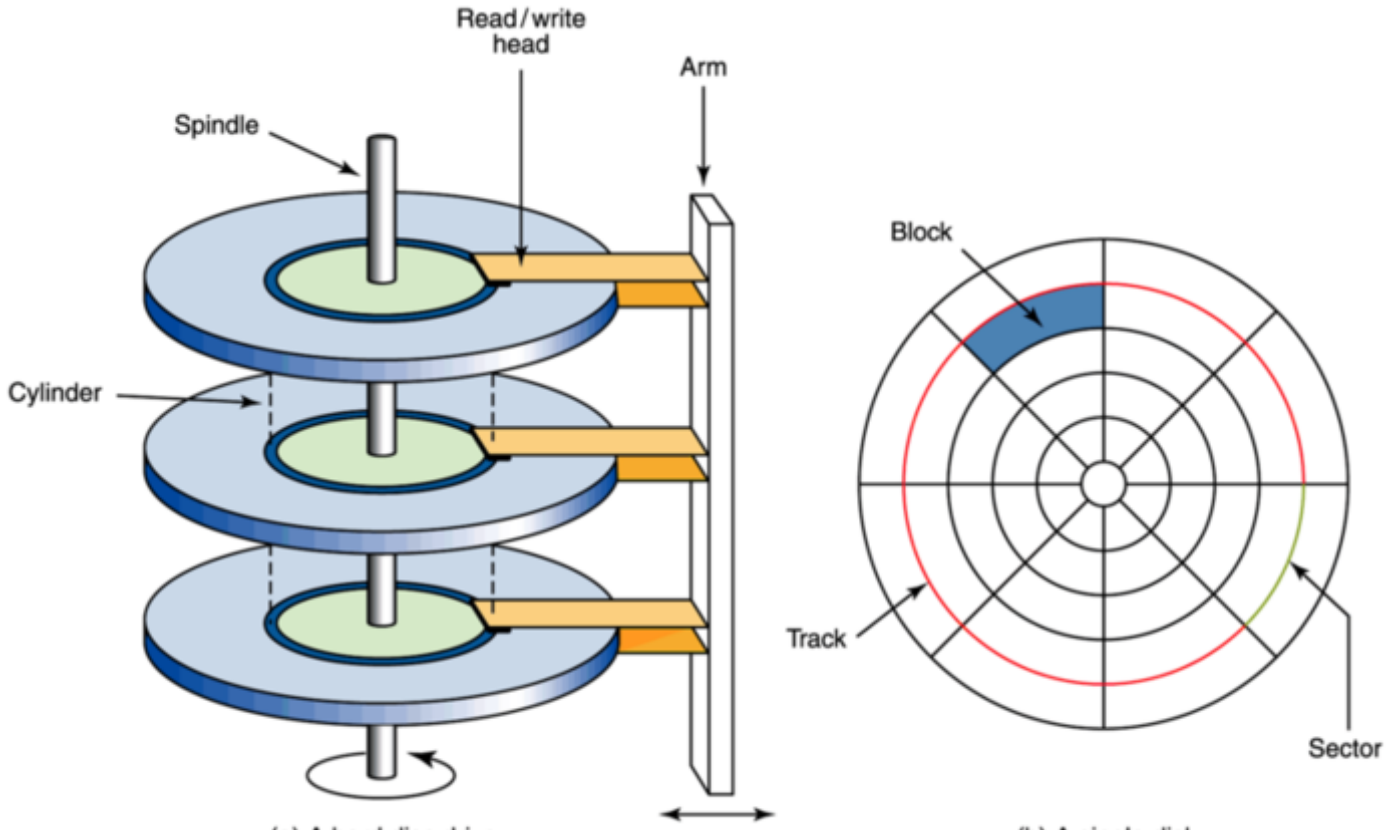
- Hard Disk is sealed unit. It contain one or more circular **platters**.
- Platter contains magnetic coating on both of its surface. Platters are stacked one on top of another.
- They rotate together round a central **spindle.**
- Rotation speed is Normally – 3600, 5400, 7200 rpm.
- It has 1 to 8 platter. Each plate has two surface.
- **Read/Write Head** used to read and write data on/f
- As many surfaces, those many R/W heads



- A platter is considered to be a collection of circular, concentric, flat rings.
- These rings are called ***TRACKS***.
- Data is stored in form of bits (0 or 1) .
- More than thousands of tracks on a single surface.
- Each track is divided into fixed size blocks, called sectors. Normally size of sector is ***512 bytes***.
- And also hundred of sectors on a single track.
- Sector is a smallest physical storage unit on disk.
- Group of tracks with similar radius are called ***cylinders***.
- No of cylinders= no of tracks


track t
spindle
arm assembly
sector s
cylinder c
read-write
head
platter
arm
rotation

Logical Structure



Logical Structure

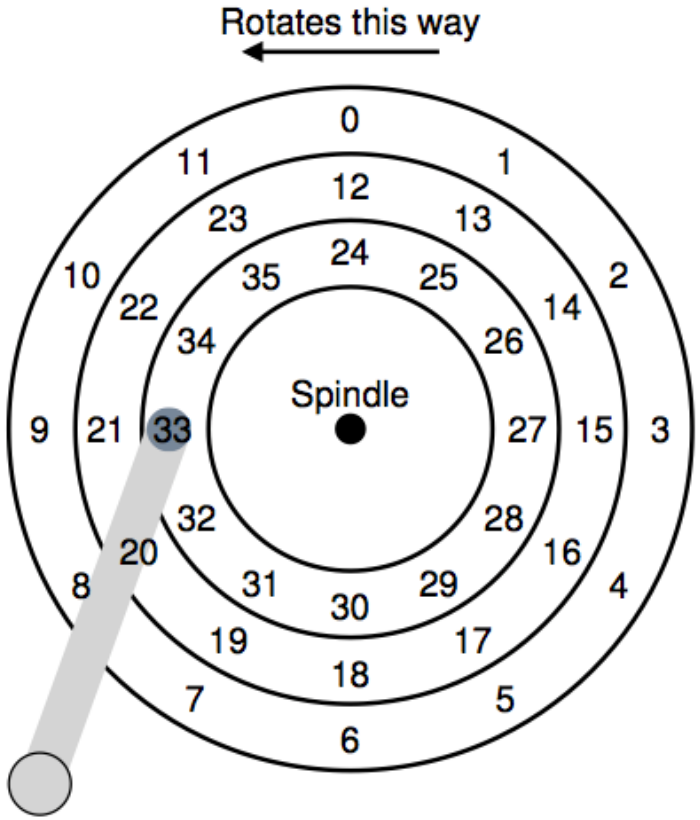
- Disk is considered as a large one dimensional array of fixed size block. Size of block is varied from system to system. (i.e. 512 bytes).
- Each block has identification number (i.e. 0,1,2,3….N-1). Total no of blocks depends upon a storage capacity of a disk.
- When info needed system generates logical block number which is converted into equivalent physical cylinder number, head number, sector number.

Addressing (Two Methods)

1. **CHS (Cylinder-Head-Sector)**
   - Logical block number generated by system is converted into equivalent Cylinder number head number and sector number.
   - Cylinders are numbered from 0 to some maximum and sectors of tracks are also numbered from 0 to some maximum
   - Read/write heads are also numbered hence the combination of CHS identifies unique sectors on disk.

2. LBA (Logical Block Addressing)

- This method identifies sectors by simply specifying sector number.
- Each sector assigned with specific sector number starting from 0.
- Disk is considered as a large one dimensio array of fixed size logical blocks.
- Sector 0 is the first sector of the first track on t outermost cylinder. Then it follows outer inner track.
- First , all sectors on that track are numbered th other tracks in that cylinder are covered a then the rest of the cylinders are covered order from outermost to innermost.



Disk I/O

- Disk I/O refers to read/Write data stored on disk. Data is read from Disk and write into Disk.
- HDD is collection of sectors. Sector is smallest unit of data transfer. (i.e. 512 bytes)
- To perform Read and Write OS needs **locations – from where to read/write.**
- **Hard Disk Controller –** is an interface between OS and Disk. Which converts different OS commands into proper signal. (i.e. LBA to CHS)
- Then it will do following –

1. **Move Disk arm to positions of head at specified cylinder. (Time = Seek Time)**
2. **Wait till specified sector is directly comes above / bellow Head (Time = Rational Latency)**
3. **Read/Write data**

File…

- *" A file is named collection of related information. Which is stored in secondary Storage"*
- Stores data Permanently and suitable for large information.
- Stores info like – **text, audio, video, image, database tables, machine code** etc.
- File is an independent entity from process, users and machines. It remains as it is when
  - Process terminates, User(owner) of file logs off system, Machine (Computer) turn off.
- Information from file can be shared among processes, users and machines.
- File is stored on CD, DVD, HDD, USB, Magnetic tapes etc.
- File is Static/ Passive entity. It has longer lifespan.

File System

- *"A part of an operating system which deals with files, is known as file system or file manager"*

- **File manager –**
  - Manages all the files in system.
  - It provides facility to **create, delete and update** file.
  - It provides **directories (folder)** to organize file.
  - It provides **efficient access** to information stored in file.
  - It **manage memory space** on secondary storage. **(not Main memory) when files are created, modified or deleted.**
  - It provides **protection mechanism** to allow users to administer how other user can access information of file.

Why we need files?

- All computer applications need to store and retrieve information. While a process is running, it can store a limited amount of information within its own address space.
- **This may lead to some problem.**
    1. **Limited storage capacity. (But large application like Banking system need large data)**
    2. **Data is lost while process terminate. (But we need data for longer time)**
    3. **Multiple process can't access data from LAS of process.**
- The usual solution to all these problems is to store information on disks and other external media in units called ***files.***
- Processes can then read them and write new ones, create, delete and update a file.
- Part of the operating system dealing with files is known as the file system

User view of File system

- File can be viewed from two point of view **User** and **System.**
- **User's Point of View (Primary goal is user convenience)**
  - How file **appears** to the user.
  - How file are **named.**
  - How file are **protected.**
  - What **operations** are allowed on file**.**
- **System point of view (Primary Goal is Efficiency)**
  - How file are **managed.**
  - How file are **accessed.**
  - How **memory is allocated** to file.

File naming

- Files are stored on Disk. This information need to be read /write again and again. So we must give **name** (for identification) so we can refer this file letter on.
- Name may be **255 character long**.
- Example.
  - ✓ Test
  - ✓ Xyz123
  - ✓ Chap_1
- File name has two part separated by (.) **[i.e. test.c]**
- **Syntax : filename.extension**

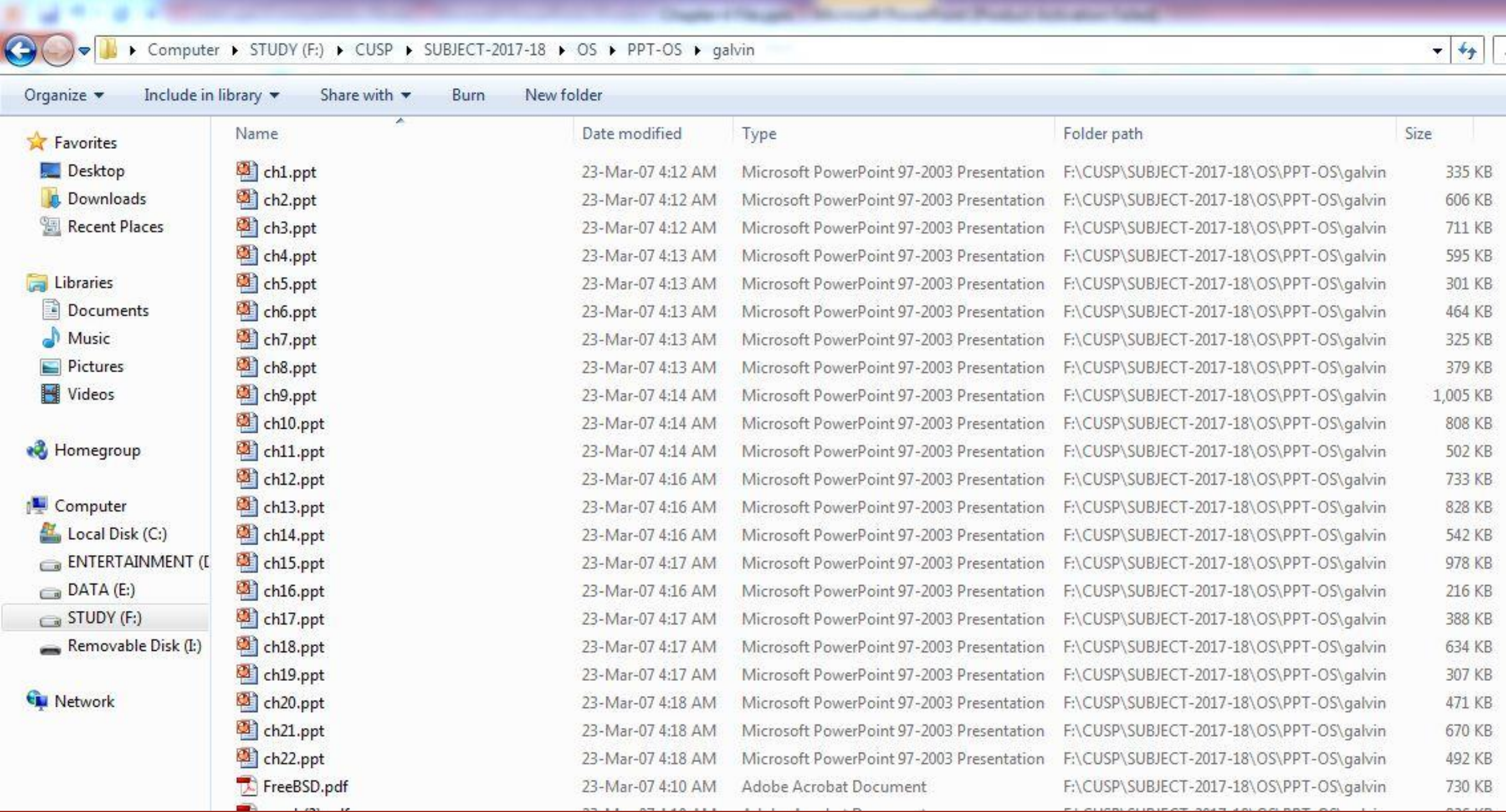
Extension	Meaning
.txt	General Text File
.c	C Source File
.bak	Backup File
.obj	Object File (Compiler output but yet not linked)
.exe	Executable File
.gif	Graphical Interchange Format File
.hlp	Help File
.html	Hyper Text Markup Language File
.jpg	Still picture encoded with JPEG format
.mp3	Music encoded in MPEG Layer 3 audio format
.mpg	Movie encoded with MPEG standard
.pdf	Portable Document Format file
.ps	PostScript file
.zip	Compressed archive file

File Attributes

- **Name – only information kept in human-readable form**
  - A String of alpha-numeric character and special symbol like (_) . Used by user to refer file
- **Identifier – Unique Tag of file**
  - Uniquely identifies the file within the file system. Used by OS to refer file.
- **Type – needed for systems that support different types**
  - it is generally expressed in the form of a file extension. For example, prog.cpp to indicate C++ file
- **Location – pointer to file location on device**
  - Pointer to the device and location on the device of the file

File Attributes

- **Size – current file size**
  - Specifies the size of file in bytes, word or blocks. And also maximum allowed size of file
- **Protection –**
  - It determines who can read file, write file, execute file and so on.
- **Usage count –**
  - Indicates number of processes that are currently using this file.
- **Time, date, and user identification –**
  - Data for protection, security, and usage monitoring Information about files are kept in the directory structure, which is maintained on the disk.
  - It specifies information regarding file creation, update and last access.


Computer ▸ STUDY (F:) ▸ CUSP ▸ SUBJECT-2017-18 ▸ OS ▸ PPT-OS ▸ galvin
Organize
Include in library
Share with
Burn
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Homegroup
Computer
Local Disk (C:)
ENTERTAINMENT (D
DATA (E:)
STUDY (F:)
Removable Disk (I:)
Network
Name
Date modified
Type
Folder path
Size
ch1.ppt 23-Mar-07 4:12 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 335 KB
ch2.ppt 23-Mar-07 4:12 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 606 KB
ch3.ppt 23-Mar-07 4:12 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 711 KB
ch4.ppt 23-Mar-07 4:13 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 595 KB
ch5.ppt 23-Mar-07 4:13 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 301 KB
ch6.ppt 23-Mar-07 4:13 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 464 KB
ch7.ppt 23-Mar-07 4:13 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 325 KB
ch8.ppt 23-Mar-07 4:13 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 379 KB
ch9.ppt 23-Mar-07 4:14 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 1,005 KB
ch10.ppt 23-Mar-07 4:14 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 808 KB
ch11.ppt 23-Mar-07 4:14 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 502 KB
ch12.ppt 23-Mar-07 4:16 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 733 KB
ch13.ppt 23-Mar-07 4:16 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 828 KB
ch14.ppt 23-Mar-07 4:16 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 542 KB
ch15.ppt 23-Mar-07 4:17 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 978 KB
ch16.ppt 23-Mar-07 4:16 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 216 KB
ch17.ppt 23-Mar-07 4:17 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 388 KB
ch18.ppt 23-Mar-07 4:17 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 634 KB
ch19.ppt 23-Mar-07 4:17 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 307 KB
ch20.ppt 23-Mar-07 4:18 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 471 KB
ch21.ppt 23-Mar-07 4:18 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 670 KB
ch22.ppt 23-Mar-07 4:18 AM Microsoft PowerPoint 97-2003 Presentation F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 492 KB
FreeBSD.pdf 23-Mar-07 4:10 AM Adobe Acrobat Document F:\CUSP\SUBJECT-2017-18\OS\PPT-OS\galvin 730 KB

File Operation

- Operating System provides **system call** to perform different operations on file. This system call can be **called using library function** provided by programming language.
- The basic operation are….

1. **Create : -**
   - ✓ Necessary disk storage space is allocated to file.
   - ✓ An entry for new created file is made in the directory. This contains file name, id number etc.
2. **Delete: -**
   - ✓ When file is no longer needed it must be deleted to free up Disk space. While deleting File space on disk is released.
   - ✓ Directory entry is deleted.

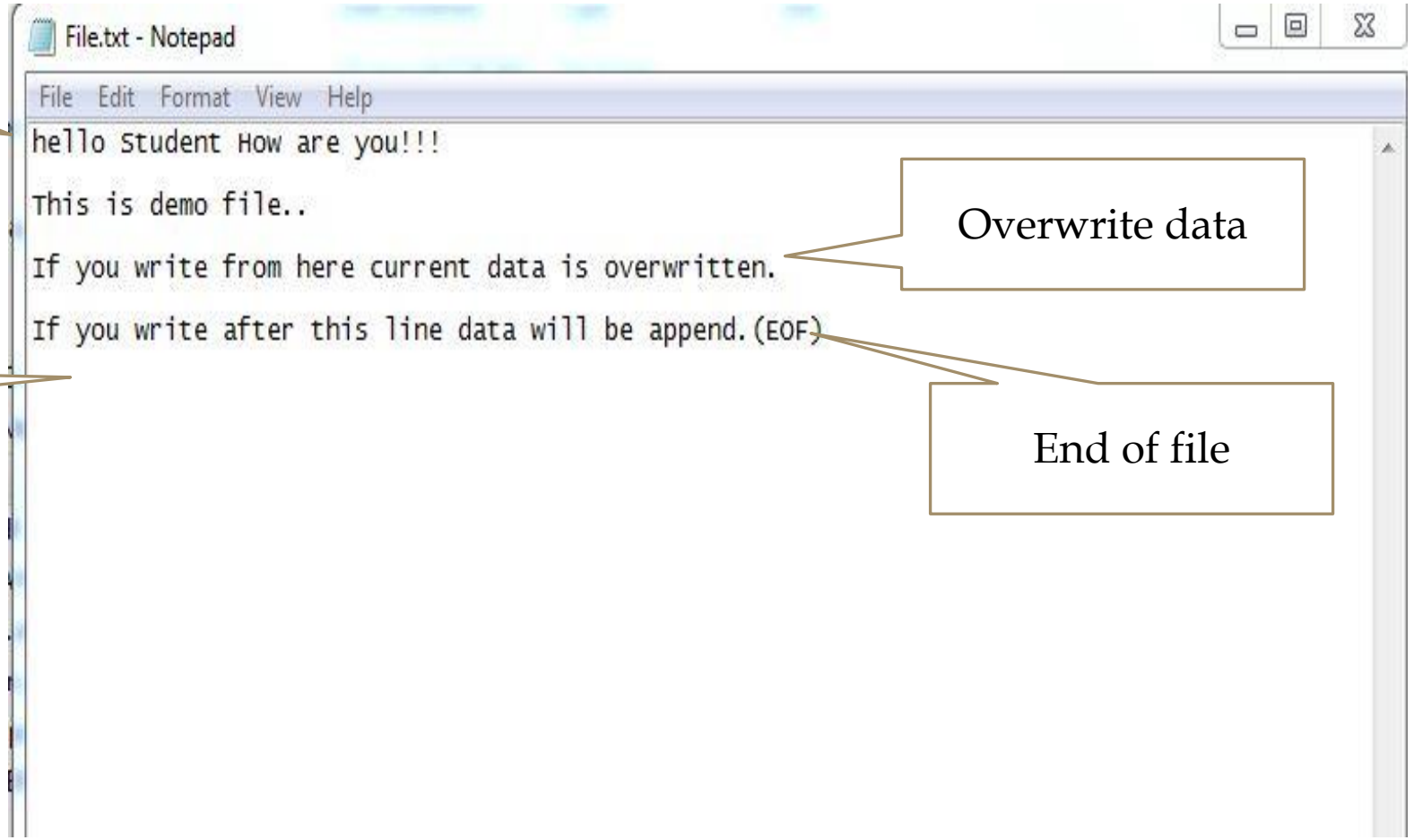
File Operation

3. **Open : -**
   - ✓ Before using file it must be opened.
   - ✓ File attributes and data are loaded into main memory.
4. **Close : -**
   - ✓ After use of file is completed it must be closed to free up main memory.
   - ✓ File attributes and data are stored back to Disk.
5. **Read : -**
   - ✓ System maintains a read pointer. It specifies location in a file from where to read data content. At beginning pointer is at starting location.
   - ✓ User needs to specify info like file name, how much to be read, and where to put that data

File Operation

6. **Write : -**
   - ✓ System maintains write pointer to specify location in the file where to write data.
   - ✓ If data is written at the end of file, size is increased and if data is written at the middle of file current data is overwritten and lost.
7. **Append : -**
   - ✓ Data are only added at the end of file.
8. **Seek :-**
   - ✓ Read/Write data from specific location. It repositions read/write pointer according to need.
9. **Get Attribute and Set attribute :-**
   - ✓ This operations are to get and set file attributes.
10. **Rename :-**
    - ✓ Rename a file.(change file name)

Read/Write Pointer will be at this position in beginning



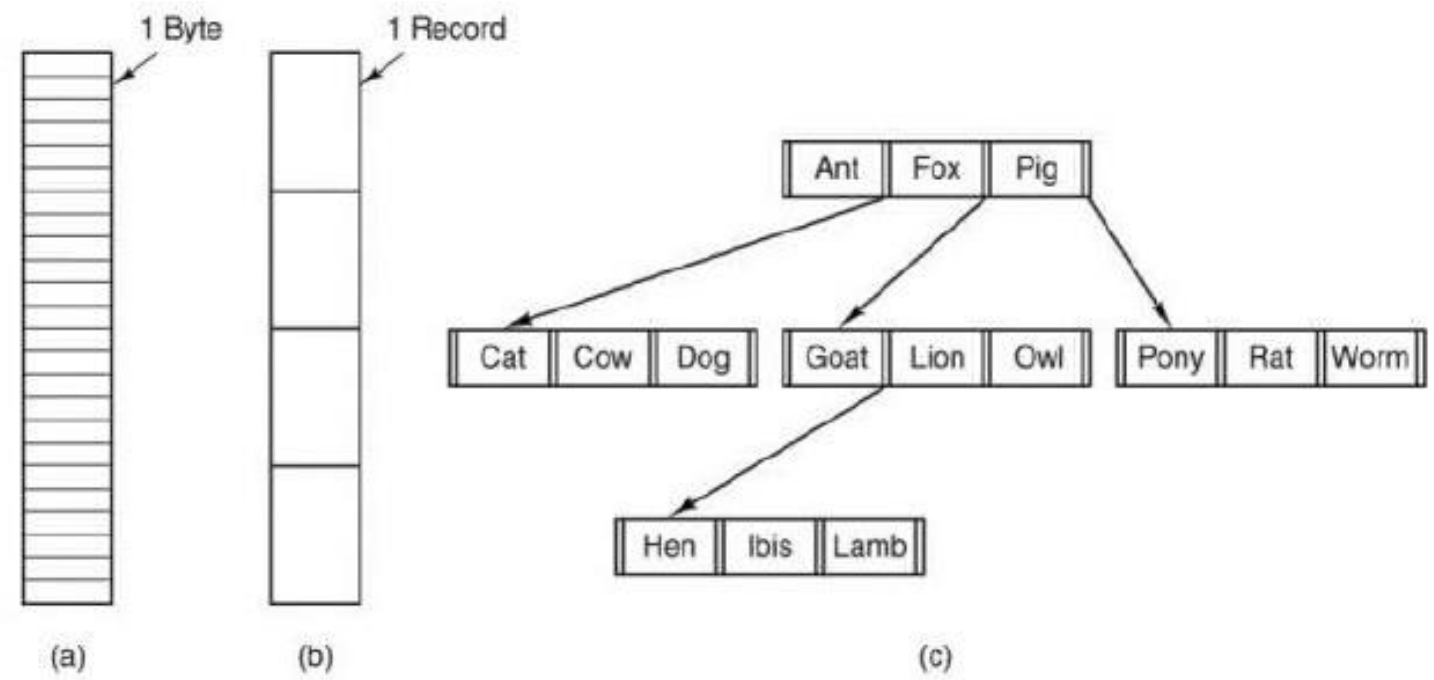
Overwrite data

Append data

End of file

File Structure

- Definition : " *File structure is the way of storing data contents in the file.*"
- It represents how data is stored in the file.
- 1. Byte Sequence 2. Record Sequence 3. Tree Sequence



File Structure

1. **Byte Sequence**
   - ✓ Data are stored as unstructured sequence of bytes.
   - ✓ OS doesn't care what is in the file. File is treated as an array of bytes.
   - ✓ Read operation returns one byte and write operation writes/append one bytes.
   - ✓ Both windows and UNIX use this approach.
2. **Record Sequence**
   - ✓ Data are stored as a sequence of **fixed length record**. Each record have its own structure. Read/Write operation access (reads or writes) one record.
   - ✓ Suitable for database applications.
3. **Tree**
   - ✓ Data are stored as a tree of **variable length records**. Each record has key in fixed position in record.
   - ✓ Record are sorted by key for easy search and retrievals.
   - ✓ Used with high mainframe systems.

File Types

- **Regular File**

  Contains User information.

- **Directories**

  They are container for other file and other sub directories.

  used to manage other files and sub directories.

- **Character Special files**

  They are related to I/O. Used to model serial I/O devices such as Terminal, printers.

- **Block Special Files**

  They are also related to I/O. Used to model block I/O devices such as Disks.

File Type (types of regular files)

Three ways to differentiate among such types of category.

- **Encode type into file *Extension***
  - ✓ File extensions are used for different types of file.
  - ✓ For bitmap image file extension will be **.bmp**
- **Encode type into file *Attribute***
  - ✓ Each file contains attribute called **file type.**
- **Encode type into file *Data***
  - ✓ File type also can be encoded in file data.
  - ✓ In UNIX, a magic number is used to identify executable file. It contains the first few bytes of Data

Directory

- Now a days capacity of secondary storage is in GBs and TBs.
- To manage this larger information in File we need to organize them in well manner.
  1. Disk are splits into one or more **Partitions** known as **Drives** (i.e ‘C’ , ‘D’ ‘E’) in windows environments.
  2. Each partitions contains various **Directories** or **Folder** to group files and sub-directories

Directory

- **“ A directory is container for different files and sub directories.”**
- In GUI based system it is called as Folder. Used to group file and sub directories.
- It provides **hierarchical structure.**
- Directory is file whose data is sequence of entry . Each entry contains file name and file attributes (i.e File Identifiers).
- **It uses two special character- Dot (.) current Directory and DotDot (..) for parent Directory.**
- **OS** uses directory entry for mapping file name with file identifiers.
- They are stored on secondary storage.

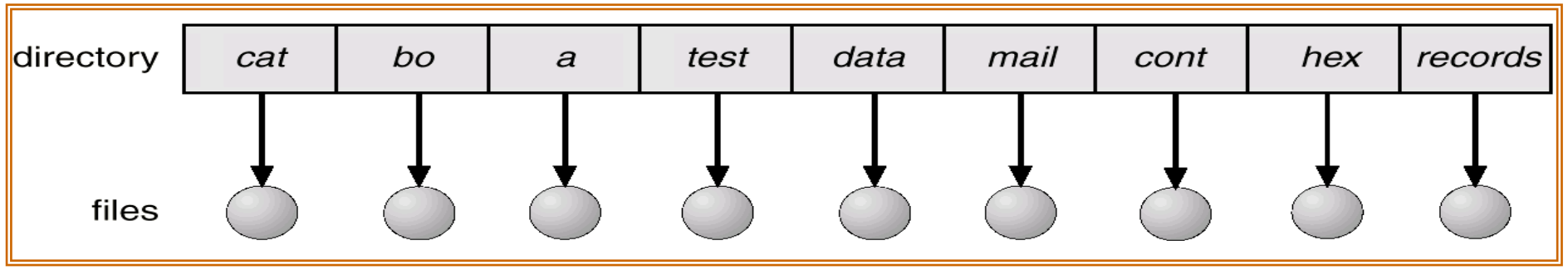
Directory Structure

- Definition:
  - " Directory structure refer to the way how directories and files are organized"
- 5 types of directory structures

1. Single level directory
2. Two level directory
3. Tree structured directory
4. Acyclic graph directory
5. General graph directory

1) Single-Level Directory

- Simplest directory structure
- Only One directory is there, it is called root directory.
- All the files are stored in that directory.
- No sub-directory.
- Remember..Names written in the rectangles are files names

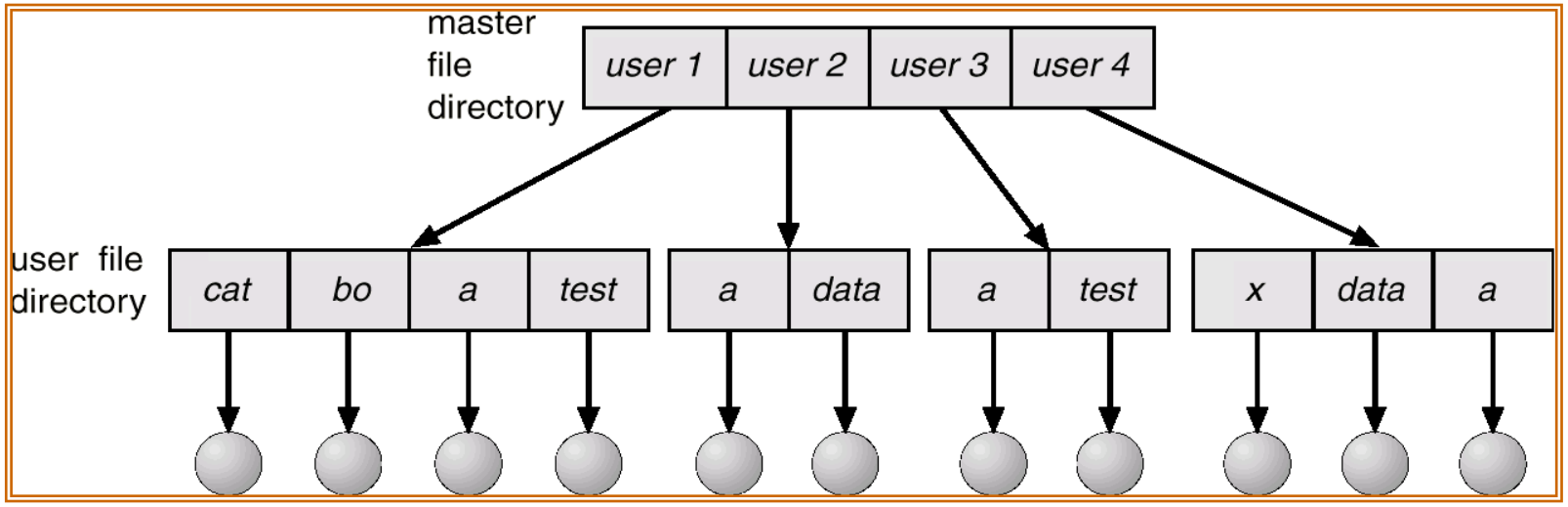


Single-Level Directory

- Advantages:
  - Simple structure
  - Searching is very fast
- Disadvantages:
  - Not suitable for multi-user systems (problem of file overwritten)
  - When number of files becomes too large, user can not remember the names of all files.
  - Different files can not be grouped

2) Two-Level Directory

- Separate directory for each user
- One is root directory or master file directory (MFD) and other is user file directory(UFD)
- User directory can contain only files. They can not have sub directories.
- Can have the same file name for different users
- Efficient searching
- No grouping capability



2) Two-Level Directory

- Advantages:
  - Avoids file name collisions
  - Can be used in multi user systems
  - Searching is efficient as user has to search in its own directory
- Disadvantages:
  - Not suitable for users having large number of files.
  - One extra directory is used to store system files.
  - Different files can not be grouped

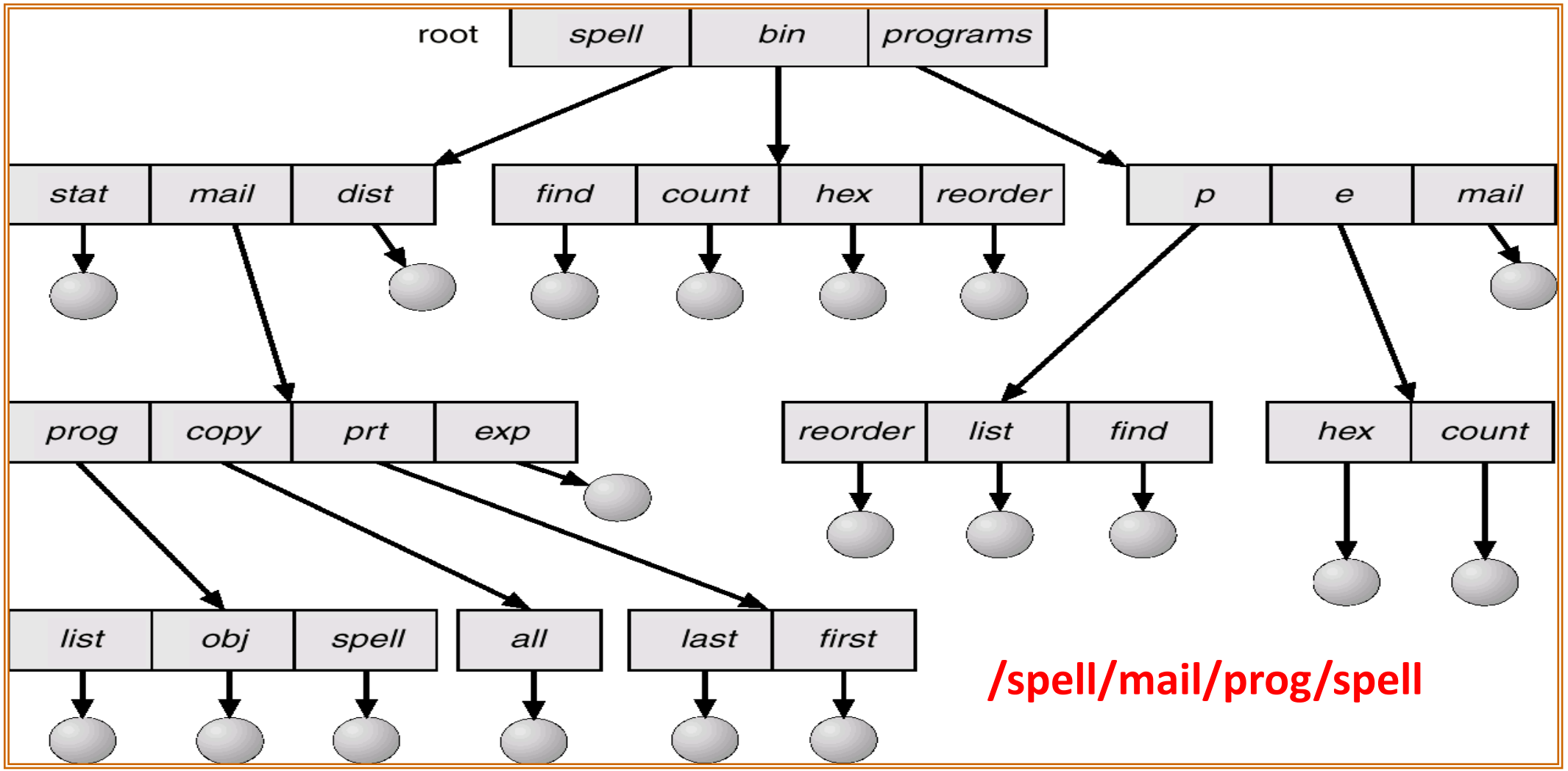
3) Tree-Structured Directories

- It Permits Users to create its own sub-directory
- also called Hierarchical File system. More than two levels of directories are possible.
- Directory can contain files as well as sub directories.
- Each file contains Unique file path.

- **Advantages**
  - Avoids file name collision
  - Can be used in multiuser system
  - Different files can be grouped

- **Disadvantages**
  - Sharing of files and directories is not possible

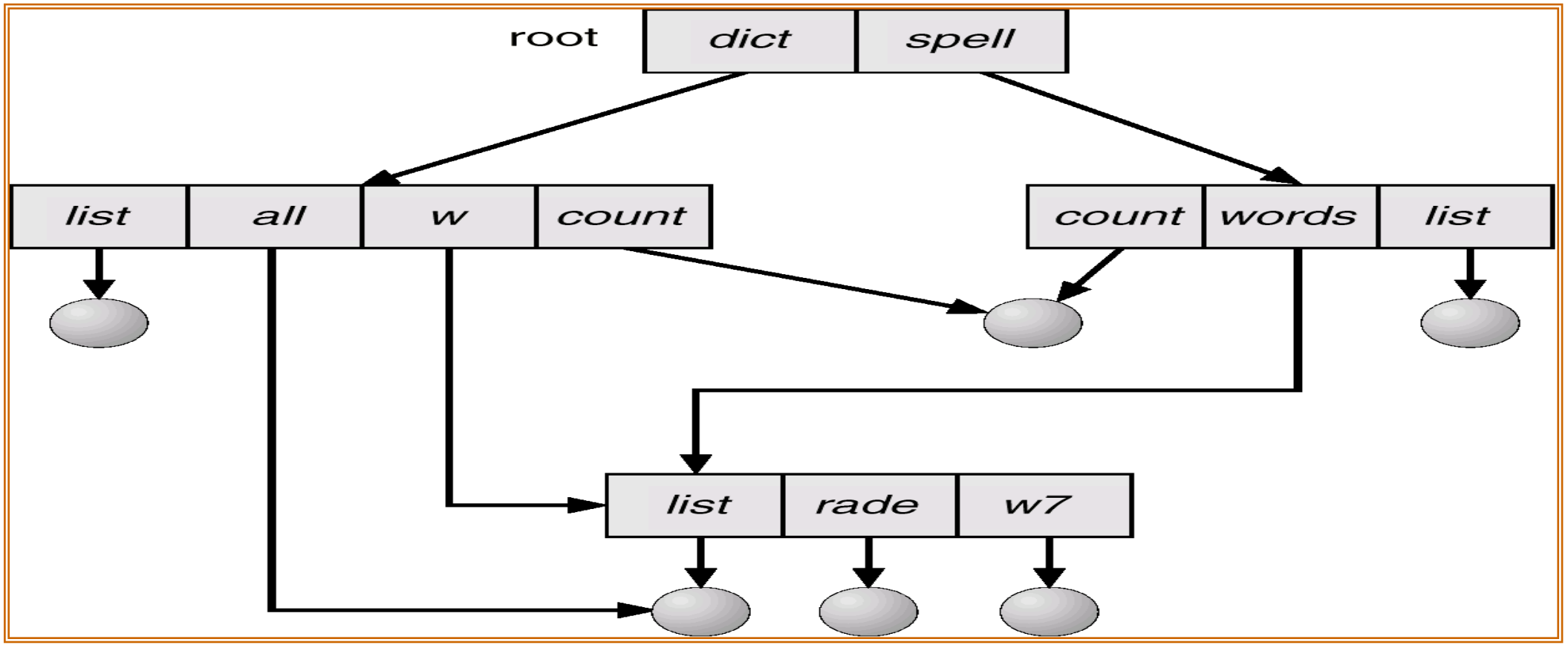
Tree-Structured Directories



4) Acyclic-Graph Directories

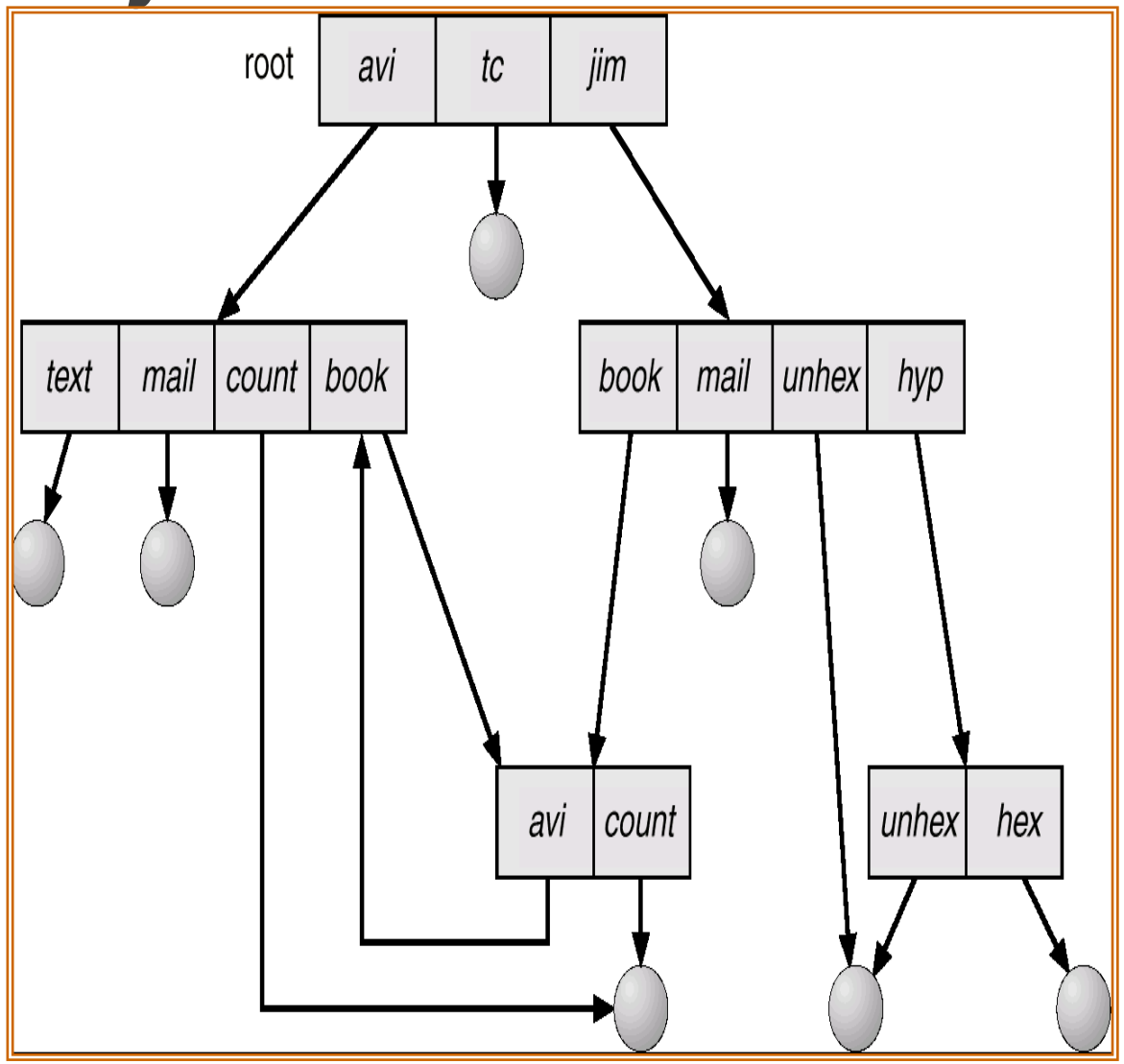
- Have shared subdirectories and files. Created using **LINK** operations.
- Two different names (aliasing). So file may contain Different file Path.
- File name 'count' is shared from two locations/directories, i.e. dict and spell.
- Directory structure does not contain Cycle.
- **Advantages**
  - Allows sharing of files and directories
- **Disadvantages**
  - Deletion of shared files or directory is complex. When file/dir is deleted, directory entry from all related directories should be deleted.
  - As files may have multiple paths, when backup is taken, multiple copies of files will be there.
  - Difficult to ensure that there are no cycles

Acyclic-Graph Directories



General Graph Directory

- It allows cycles in graph.
- **Advantages**
  - Allow Sharing of file.
- **Disadvantages**
  - Searching should be done carefully. Else it may result into infinite loop.



File Paths

- *" A file path is a character string divided into separate component by special character such as slash (/). "*
- It is used to uniquely identify a location of file or directory.
- Separate components are : Directory and file at the end of path.
- Different file may have same name but file path should be unique.
- Example.
  - ✓ C:\Progs\CPP\test.cpp …….On Windows environment
  - ✓ /user/home/test.c ……..On Unix environment

File Paths

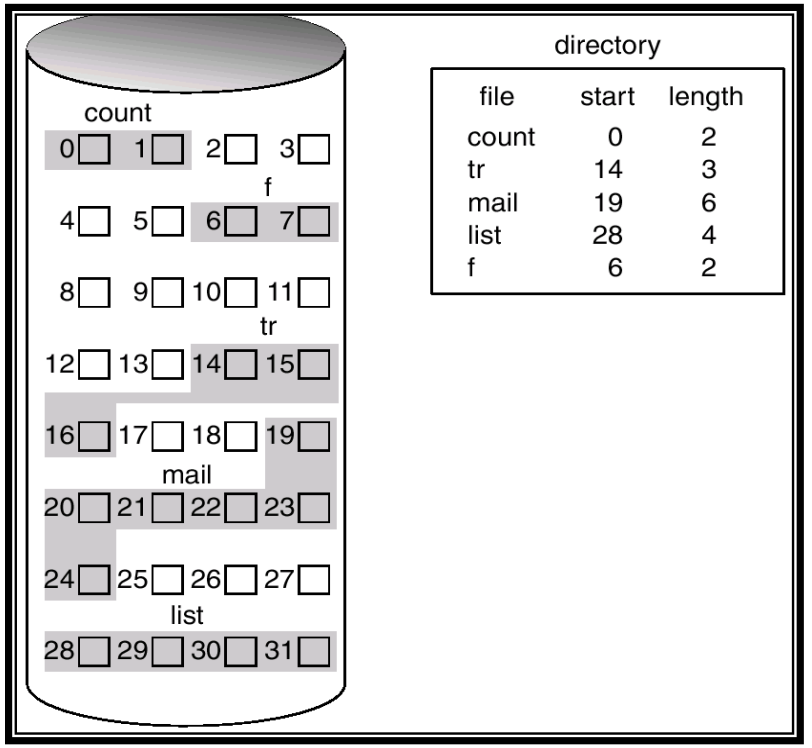
1. **Absolute File path**
   - ✓ It always start with root directory.
   - ✓ Example. C:\Progs\CPP\test.cpp and /user/home/test.c
2. **Relative File path**
   - ✓ It starts with current directory designated by users.
   - ✓ Dot (.) and DotDot(..) can be used as a current directory.
   - ✓ Example.
     A. If C:\Progs is current directory then path for file test.cpp can be given CPP\test.cpp (For Windows)
     B. If /user/home is a current directory than path for test.c file is ./test.c (For Unix/Linux)

Disk Space Allocation Methods

- Its from System points of view.
- Goals:
  - ✓ Disk Space should be utilized effectively.
  - ✓ Files should be accessed quickly.
- Files are stored on disks. Files contain data as well as attributes. Data means actual data contents stored in files and attributes contains information about files.
- Disk is considered as collection of fixed size logical blocks. General block size will be of 512 bytes, or 1KB.
- Different **Disk space allocation methods** are..
  1. **Contiguous Allocation**
  2. **Linked Allocation**
  3. **Indexed Allocation**

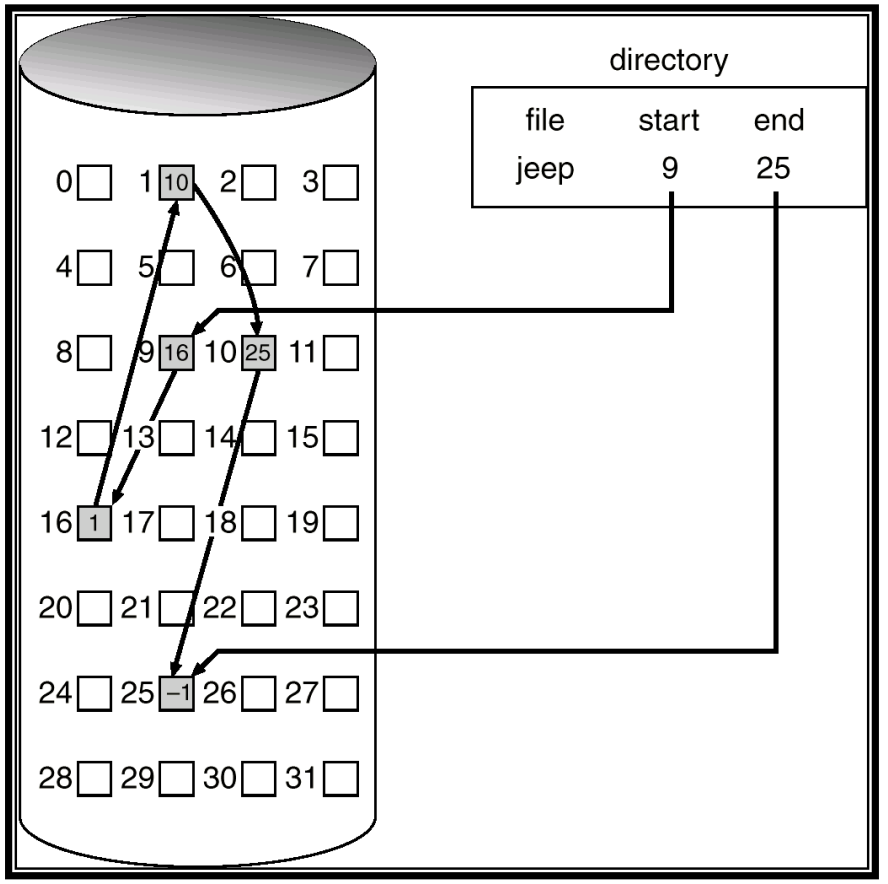
1) Contiguous Allocation

- Each file occupies set of contiguous blocks on disk.
- When file is created, a Disk is searched to find out a chunks of free memory having enough size to store a file.
- Directory entry contain File name, Starting block, and Length.
- This method is widely used in CD ROMs where the file size is known in advanced.
- **Advantages :**
  - Simple to implement.
  - File access is quick,
- **Disadvantages:**
  - Finding free block is time consuming.
  - If file size increases, accommodation of such extension is not possible.
  - External Fragmentation is possible.



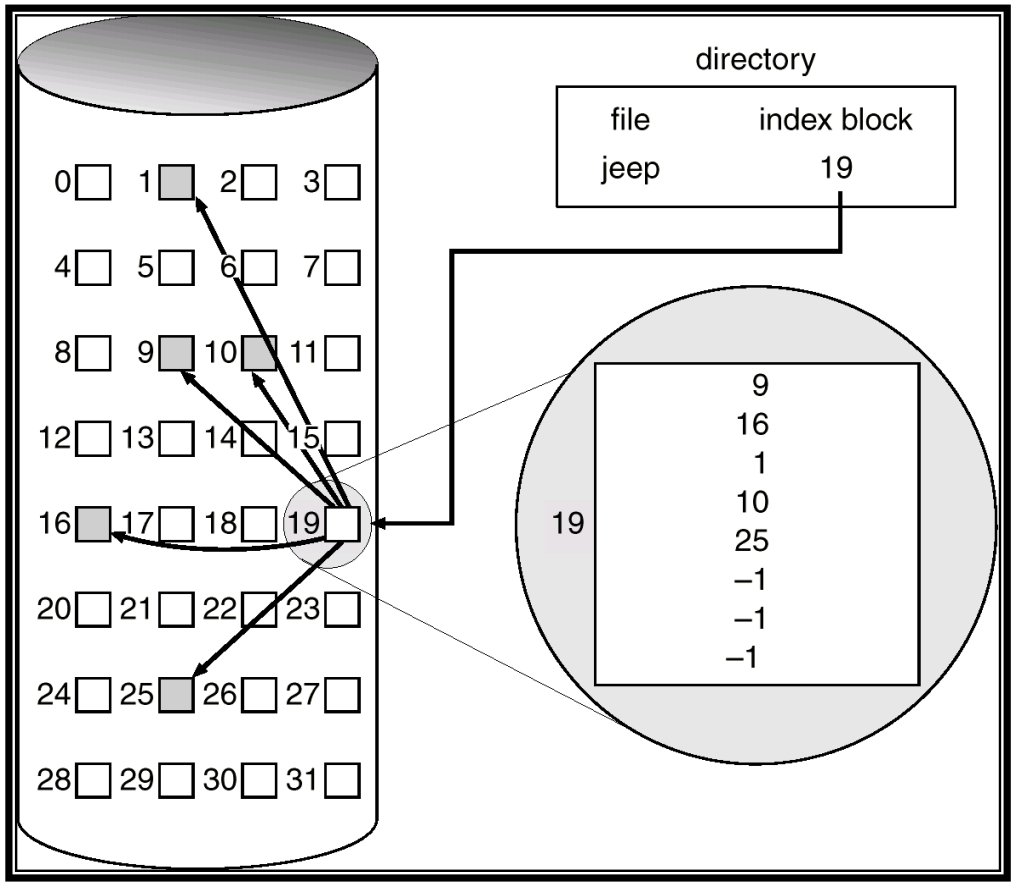
2) Linked Allocation

- Each file is a linked list of Disk Blocks. Each linked block contains pointer to the next block in a list.
- Disk Blocks may be scattered any where on Disk.
- Directory entry contain the start and last block number in linked list.
- **Advantages:**
  - Does not suffer from External fragmentation.
  - Any free disk block can be allocated to a file.
- **Disadvantages:**
  - File access is time consuming.
  - Random Access is not possible.
  - Extra space is required for pointers.



3) Indexed Allocation

- **Problem with linked allocation method:**
  - Pointers to various disk blocks are scattered on disk among various disk blocks, due to this , linked allocation can not support efficient direct access.
- It brings all the pointers together into one location: the **Index Block**
- Each file contains its own index block.
- An index block is an array of Disk Block Address. The 'I'th entry in index block points to 'I' th block.
- Advantages:
  - Does not suffer from external fragmentation
  - Direct access is efficient.
- Disadvantages:
  - Space is wasted as there are few entries in index block
  - Maximum allowable file size depends on size of an index block.



Security and Protection Mechanism

- File safety comes in two ways
  1. Reliability
  2. Protection
- **Reliability** means safety from physical damage(i.e. hardware problem, fire, dirt, power failure, head crashes).
- **Reliability** is generally provided by keeping more than one copy of files.
- **Protection** means safety from improper access.
- Protection is required where files can be shared among various users.
- **User** may want to read, write, execute its own file. He also want to share such file or programs with others. But he doesn't want other user to modify this file.
- Solution to this problem is allow limited sharing.

Security and Protection Mechanism

- This can be done by providing some access control. (i.e. **Read, Write, Execute, Append, Delete**)
- There are two most common ways to limit the types of file access.

1. **Password**
   - ✓ Password is associated with each file. Uer can access file only if he/she knows password.
   - ✓ There may be large number of files in system so user can't remember password of each and every file.
2. **Access Control**
   - ✓ Access Control List (ACL) is associated with each file. This list specifies the user name and type of access allowed for that user.
   - ✓ User type may be – Owner, Group, Others

Thank you...