SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

23CST202 – Operating Systems

II YEAR - IV SEM

UNIT 4 – FILE SYSTEMS

Syllabus

- UNIT I OVERVIEW AND PROCESS MANAGEMENT 9
- Introduction - Computer System Organization, Architecture, Operation, Process Management – Memory Management – Storage Management – Operating System – Process concept – Process scheduling – Operations on processes – Cooperating processes – Inter process communication. Threads - Multi-threading Models – Threading issues.
- UNIT II PROCESS SCHEDULING AND SYNCHRONIZATION 10
- CPU Scheduling - Scheduling criteria – Scheduling algorithms – Multiple-processor scheduling – Real time scheduling – Algorithm Evaluation. Process Synchronization - The critical-section problem – Synchronization hardware – Semaphores – Classical problems of synchronization. Deadlock - System model – Deadlock characterization – Methods for handling deadlocks – Deadlock prevention – Deadlock avoidance – Deadlock detection – Recovery from deadlock.
- UNIT III MEMORY MANAGEMENT 9
- Memory Management - Background – Swapping – Contiguous memory allocation – Paging – Segmentation – Segmentation with paging. Virtual Memory - Background – Demand paging – Process creation – Page replacement – Allocation of frames – Thrashing.
- UNIT IV FILE SYSTEMS 8
- File concept - Access methods – Directory structure – Files System Mounting – File Sharing – Protection. File System Implementation - Directory implementation – Allocation methods – Free-space management.
- UNIT V I/O SYSTEMS 9
- I/O Systems - I/O Hardware – Application I/O interface – Kernel I/O subsystem – Streams – Performance. Mass-Storage Structure: Disk scheduling – Disk management – Swap-space management – RAID – Disk attachment – Stable storage – Tertiary storage. Case study: Implementation of Distributed File system in Cloud OS / Mobile OS.

- L :45 P:0 T: 45 PERIODS

FILE SYSTEMS

- File concept
- Access methods
- Directory structure
- Files System Mounting
- File Sharing
- Protection

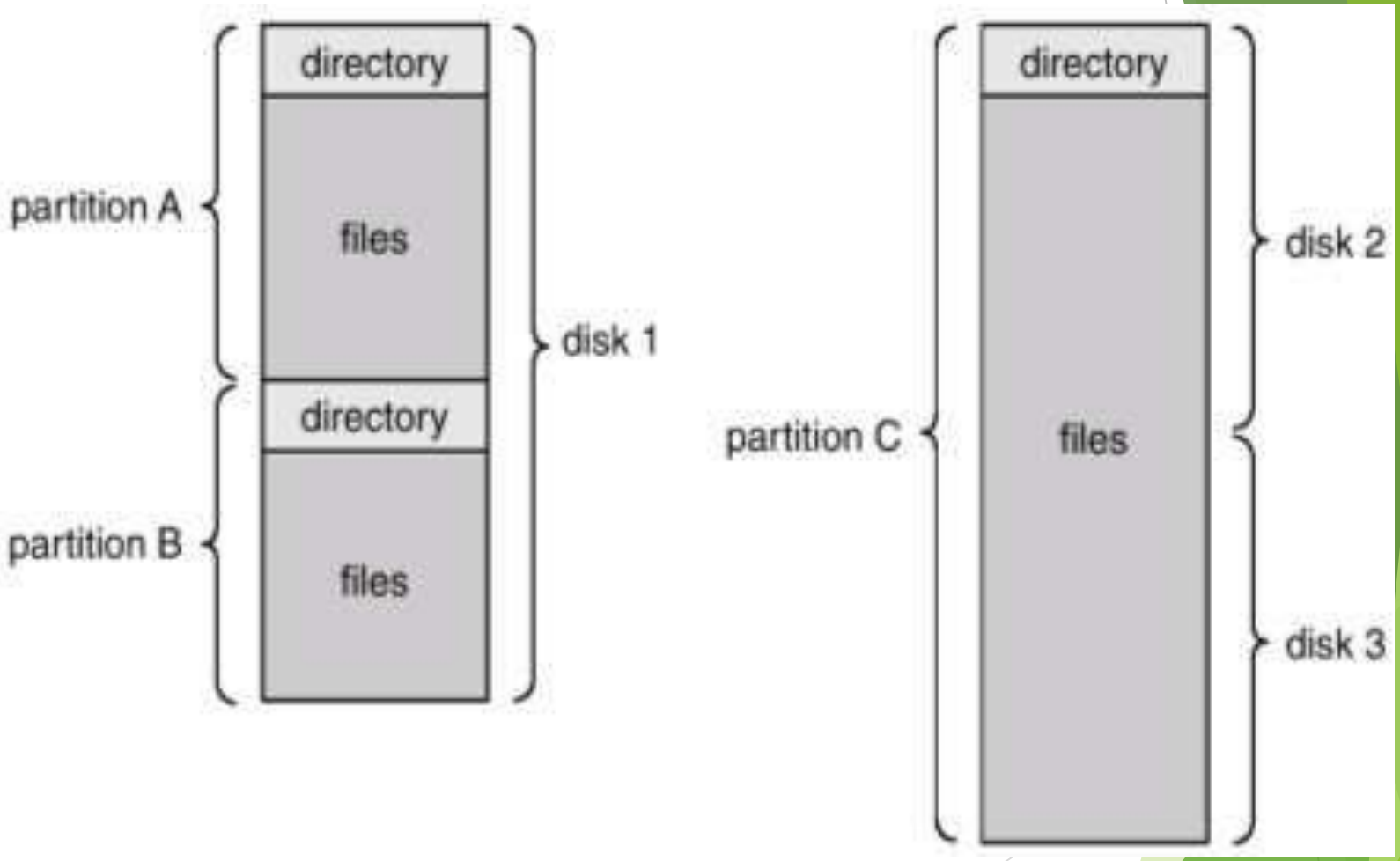
Disk Structure

- The disk which can be in the size of terabytes can be subdivided into partitions.
- A disk or partition can be used raw – without a file system, or can be formatted with a file system.
- The partitions of the disk are also known as minidisks or slices.
- Each disk has at least one partition.
- Partitions can store multiple operating systems.
- That is, each partition can have a different operating system.

Disk Structure

- Each entity containing a file system is known as a volume.
- Each volume containing a file system also needs to track that file system's information.
- In each volume, this information is maintained in a device directory or volume table of contents.
- The device directory records information such as name, location, size, type for all files in that volume.

Disk Structure

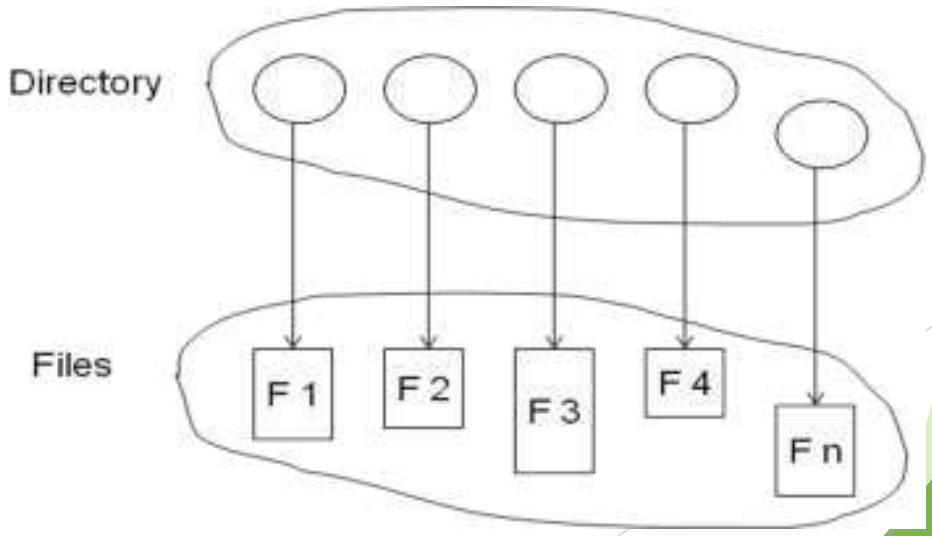


Disk Structure

- Figure shows the organization of a typical file-system.
- Disk1 has two partitions.
- Partition A has a file system and a corresponding device directory.
- Partition B has another file system and a corresponding device directory.
- It is also possible for a partition to cover two disks.
- Partition C has a file system that is kept in disk2 and disk3.

Directory Structure

- A directory structure is a collection of nodes containing information about all files that are kept in the disk.
- Both the directory structure and the files reside on the disk.
- The backups of these two structures are kept on tapes.
- Figure shows a few directories and files under the directories.



Operations Performed on Directory

- Similar to how operations can be performed on files, there are operations that can be performed on directories.
- Some of the operations that can be performed on directories are given below:

- Search for a file – A directory has information about all the files present in the directory. To search for a file, it is necessary to search the directory.

- Create a file – To create a file, an entry is created in the directory. For this, it is necessary to write into the directory.

Operations Performed on Directory

- Delete a file – To delete a file, it is necessary to remove the name of the file and all other details about the file from the directory. This again needs write permissions in the directory.

- List a directory – For listing the contents of a directory, it is necessary to read from the directory.

- Rename a file – To change the name of the file, it is necessary to read, write and search in the directory. Note that renaming a file may change the position of the file name in the directory.

- Traverse the file system – This also needs reading and searching operations on the directory.

Organizing the Directory

- It is necessary to organize a directory logically so that the following are achieved:

- **Efficiency**

  locating a file quickly.

  The directory should be organized such that files can be located quickly when searching.

Organizing the Directory

- **Naming**
  - convenient to users.
  - The names of the files cannot be arbitrary names and must be easier for the users to remember the names.
  - Two users cannot have the same name for different files.
  - The same file can have several different names.

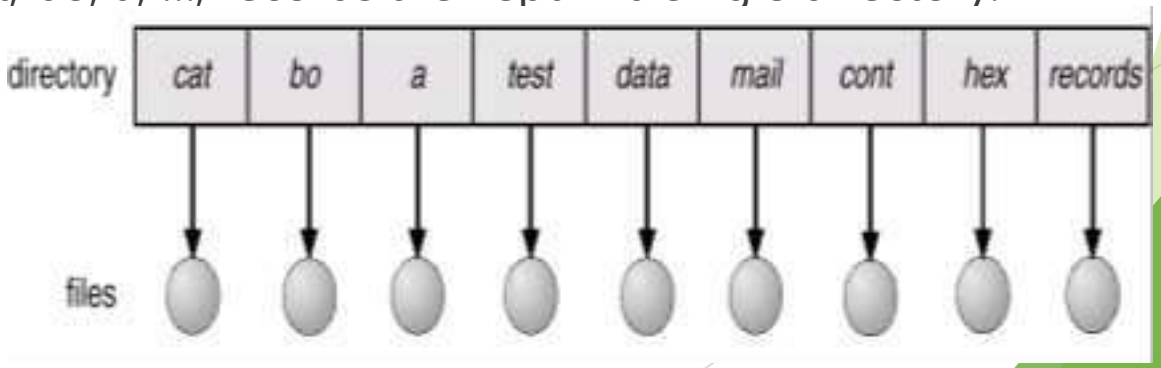
- **Grouping**
  - Based on the properties of files, it is necessary to logically group the files.
  - For example, all Java programs may have to be put under one directory, all games under another directory and so on.

Logical Structure of directory

- Single-Level Directory
- Two-Level Directory
- Tree-Structured Directories
- Acyclic-Graph Directories
- General Graph Directory

Logical Structure of directory

- **Single-Level Directory**
- In this scheme, a single directory is kept for all the users.
- This is the simplest scheme. But there are limitations in this scheme.
- Since there is only one directory, it becomes difficult to manage the files when the number of files increases.
- When the system has more than one user, it becomes difficult for different users to assign unique names to files.
- Even a single-user finds it difficult to remember the names of his/her files because all files are kept in the same directory.
- Figure shows an example of a single-level directory.
- The files cat, bo, a, …, records are kept in a single directory.



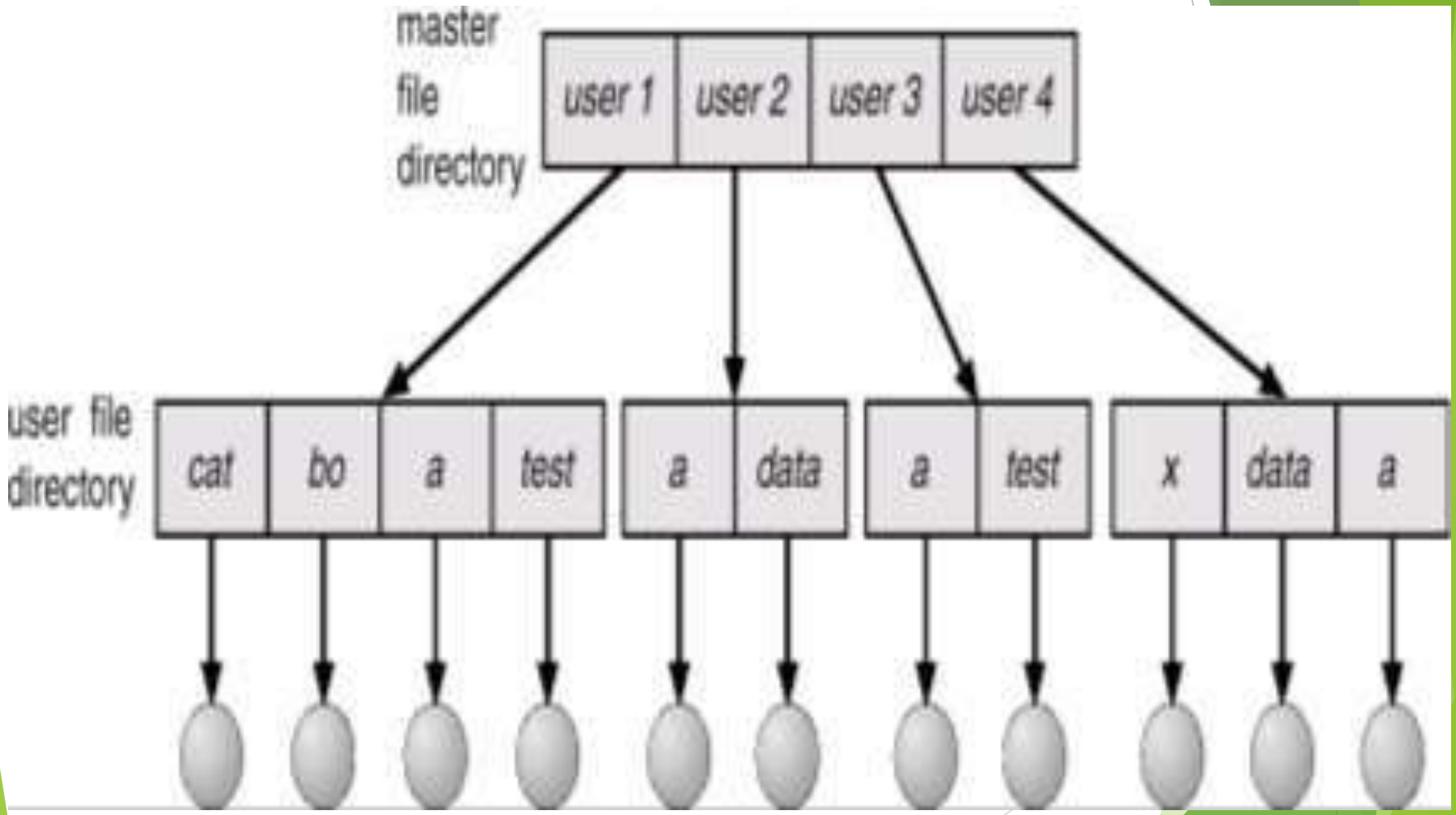
Logical Structure of directory

- **Two-Level Directory**
- In a two-level directory, there is a master file directory (MFD) and under this directory, a separate directory is assigned for each user
- This scheme has more advantages compared to the single-level directory.
- Different users can use the same file name.
- Creation and deletion of files are confined to the user's user file directory (UFD).
- Therefore, naming becomes easier in this scheme.
- But the limitation is that users can't cooperate and access one another's files.
- To access another user's file, the path name of that file should be specified.
- The path name of a file is specified using the name of the MFD (root) followed by the name of UFD and the name of file.

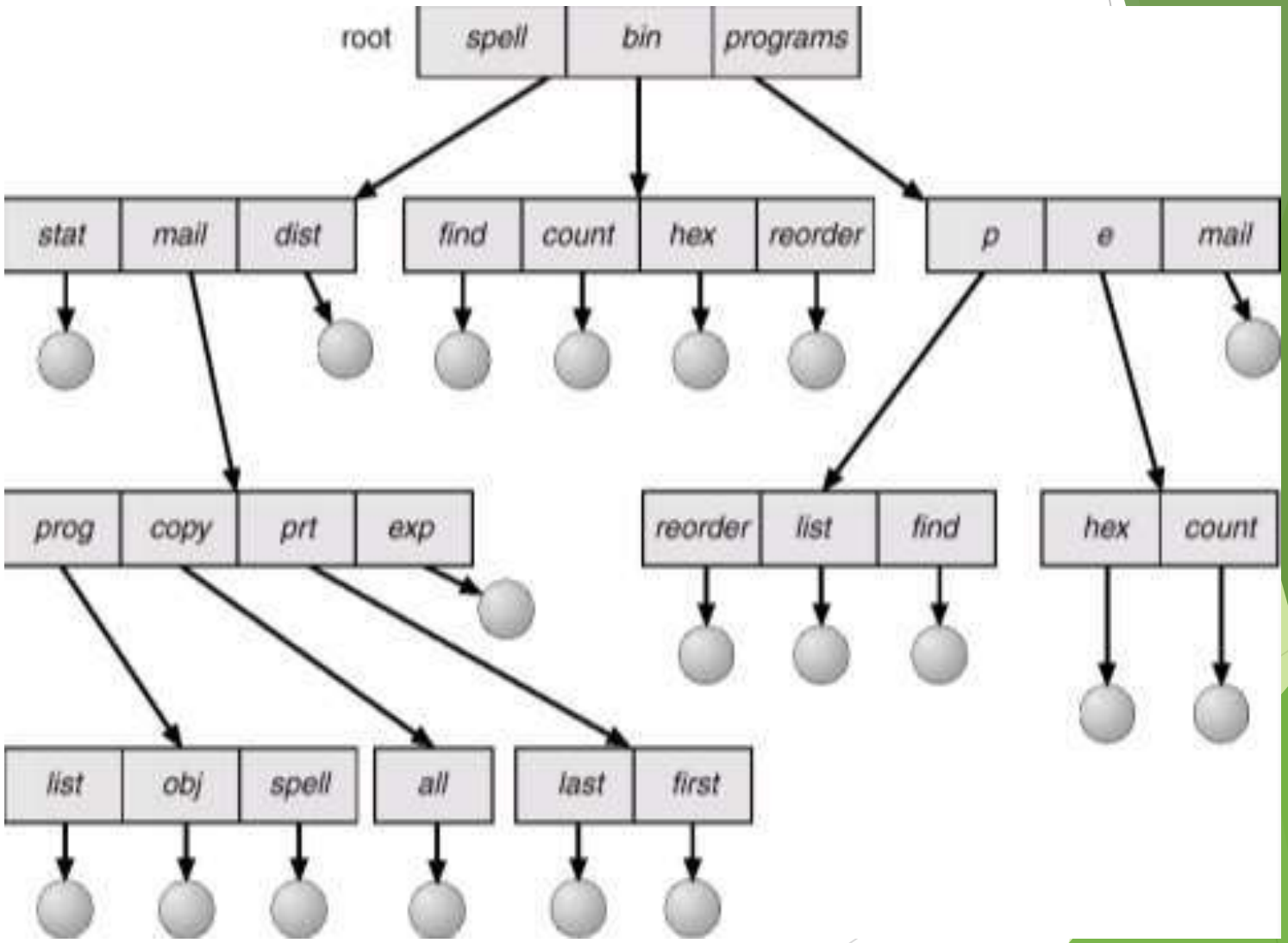
Logical Structure of directory

- Another limitation in this system is the placing of system files like loaders, assemblers and compilers.
- These files are needed by all the users.
- If these files are copied to each UFD, it is waste of space.
- Therefore, a special user directory can be created to contain the system files.
- For accessing files that are not kept in the current directory, a search path can be defined. By default, files are first searched in the current directory.
- If the files are not available in the current directory, the files are searched in the directories specified in the search path.
- The system files can also be searched in this manner.

Logical Structure of directory



Tree-Structured Directories



Tree-Structured Directories

- The two-level directory was extended to have multiple levels and the tree-structured directory was formed.
- The tree is the most common directory structure.
- Figure shows an example of a tree-structured directory.
- The directory is treated as a special kind of file.
- Under a directory, there can be files as well as other subdirectories.
- A bit is used to differentiate between a file and a directory present in the same level.
- The tree has a root directory and beneath the root directory, there are a number of subdirectories.
- System calls are available for creating and deleting directories. Every file has a unique path name.

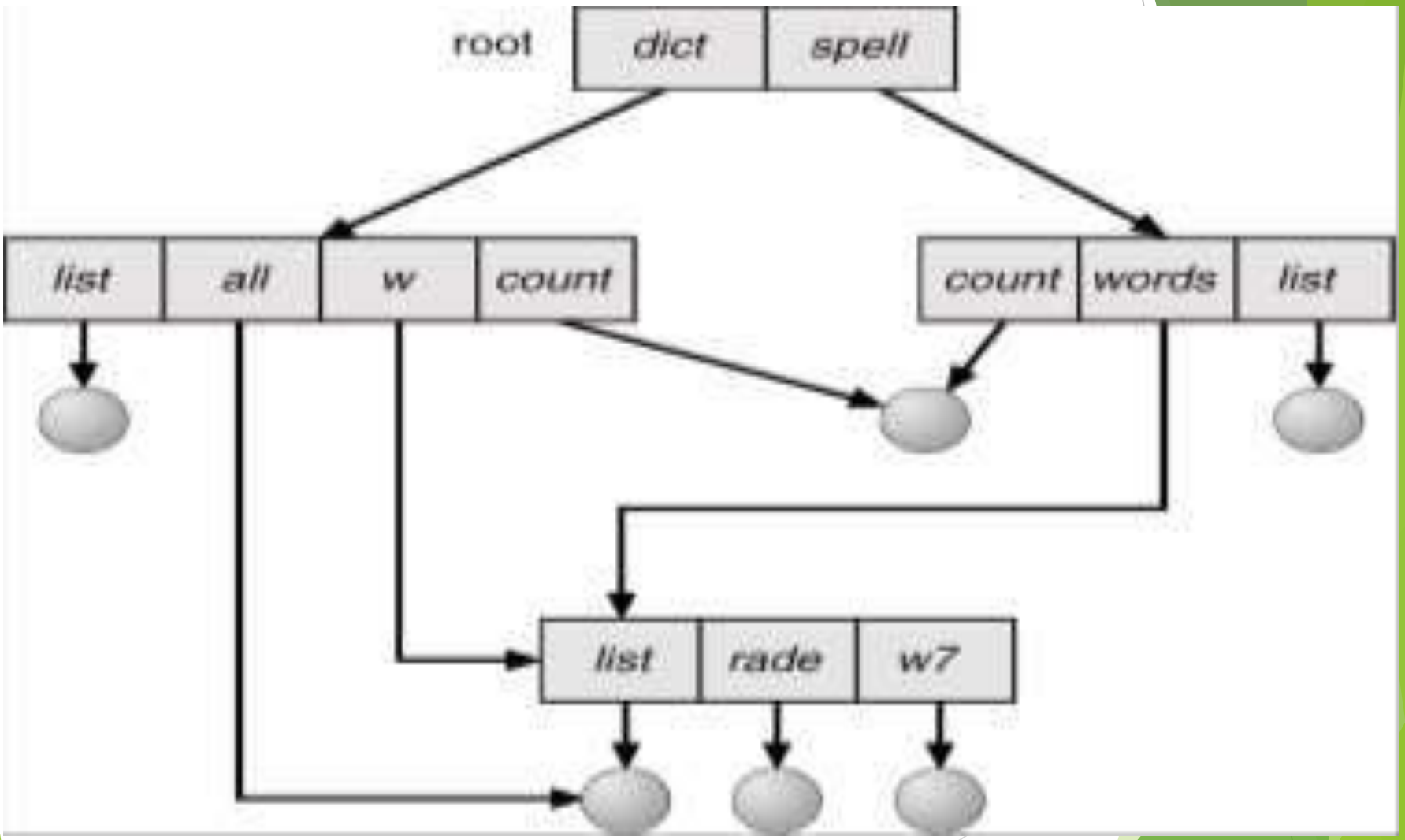
Tree-Structured Directories

- The introduction of tree-structured directories introduced two types of path names called the absolute path name and the relative path name.
- If the path name starts from the root, it is called as absolute path name.
- If the path name starts from the current directory, it is called as relative path name.
- For example, if root/spell/mail is the current directory, prt/first is the relative path name and absolute path name is root/spell/mail/prt/first.

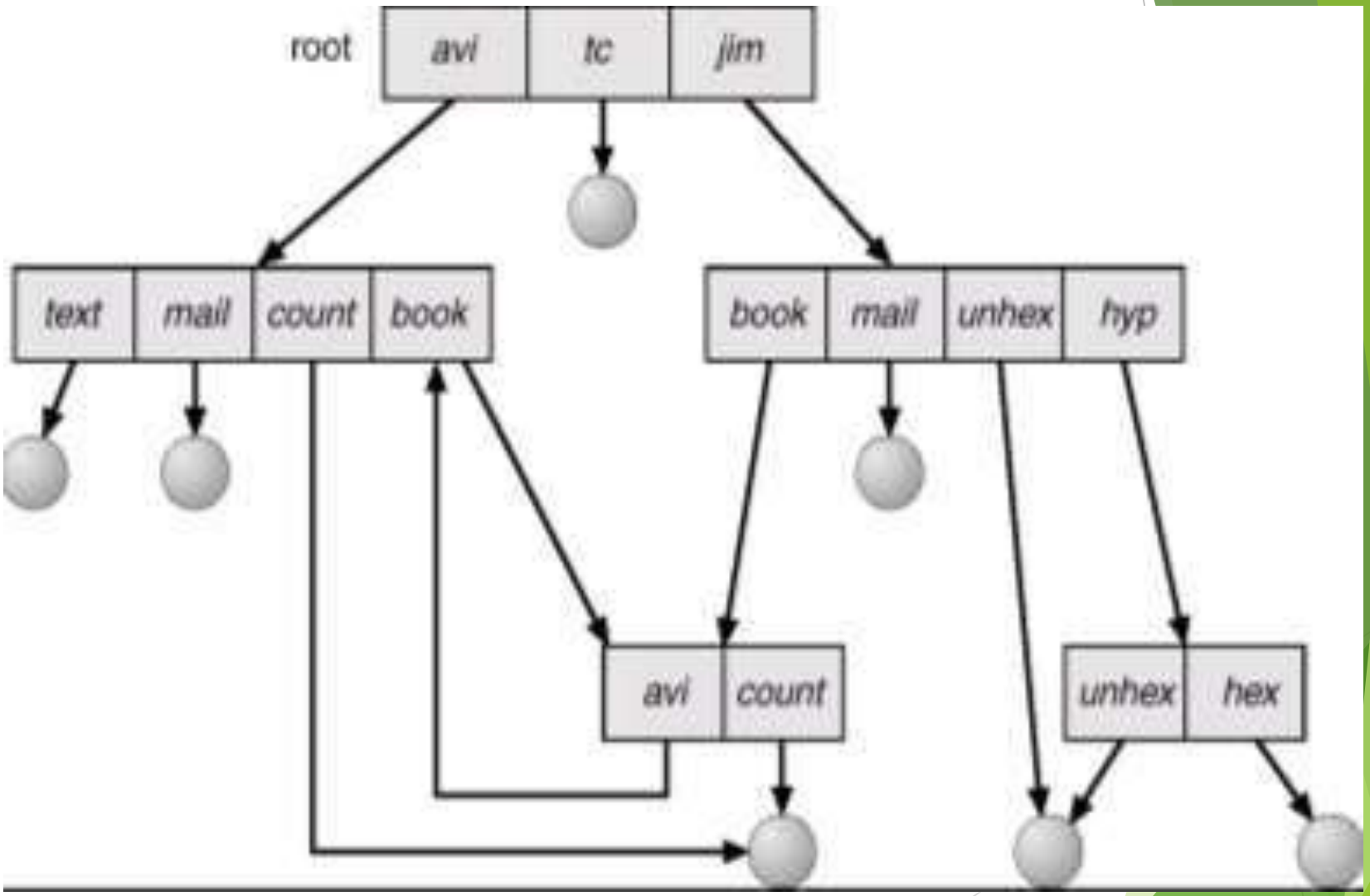
Acyclic-Graph Directories

- To overcome the disadvantage of tree-structured directories, acyclic-graph directories were formed which have shared subdirectories and files.
- Figure shows that file *count* is shared by the directory *spell* and the directory *dict*.
- Therefore, the file *count* has two different path names, /dict/count and /spell/count.
- This sharing of files is helpful when there is more than person working on a project and have to access a common file.
- This is not the same as having two copies of the same file.

Acyclic-Graph Directories



General Graph Directory



General Graph Directory

- The general graph directory is similar to the tree-structured directory except that cycles are allowed in the structure.
- A cycle is path of edges and vertices wherein a vertex is reachable from itself.
- Here, vertices denote directories or files and there is an edge between a file/subdirectory and a directory if the directory is the parent directory of the file/directory.