Department of Computer Science and Engineering

FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW
LUCKNOW

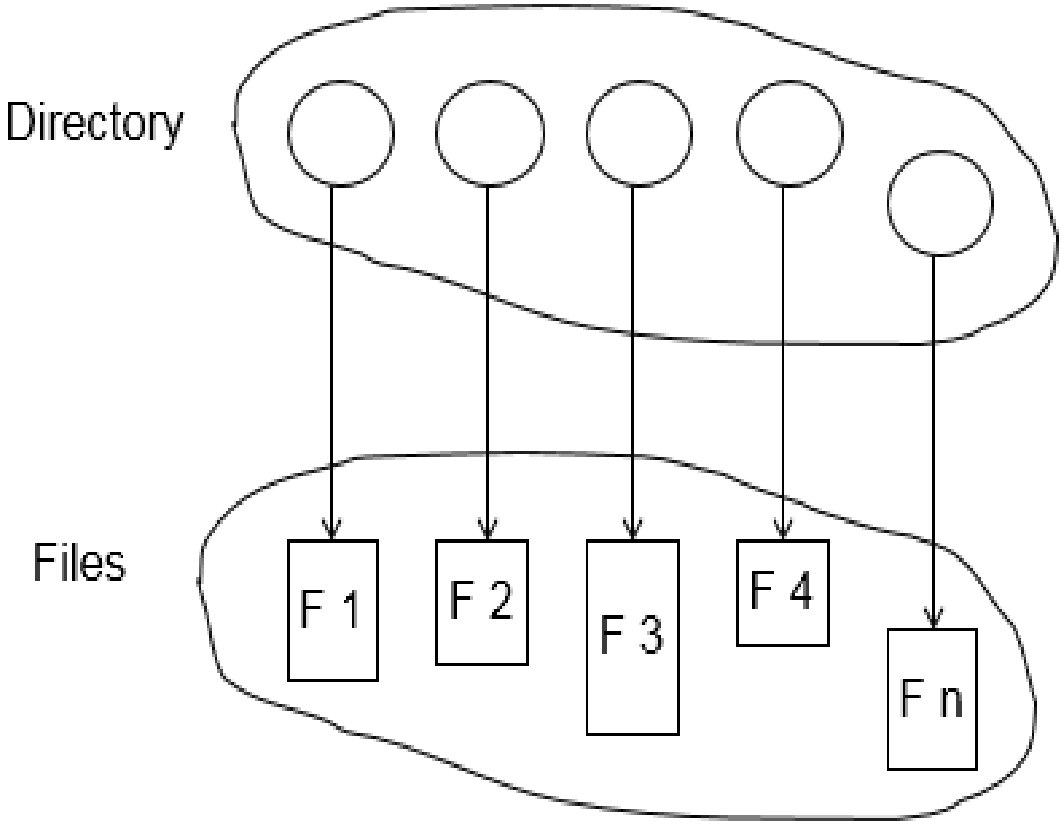
CS-501

Dr. Zeeshan Ali Siddiqui
Assistant Professor
Deptt. of C.S.E.

DIRECTORY

Directory

- The *directory* can be viewed as a *symbol table* that translates file names into their directory entries.
- Both the directory structure and the files reside on *disk*.



Directory

- **Operations Performed on Directory:**

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

Logical Structure of a Directory

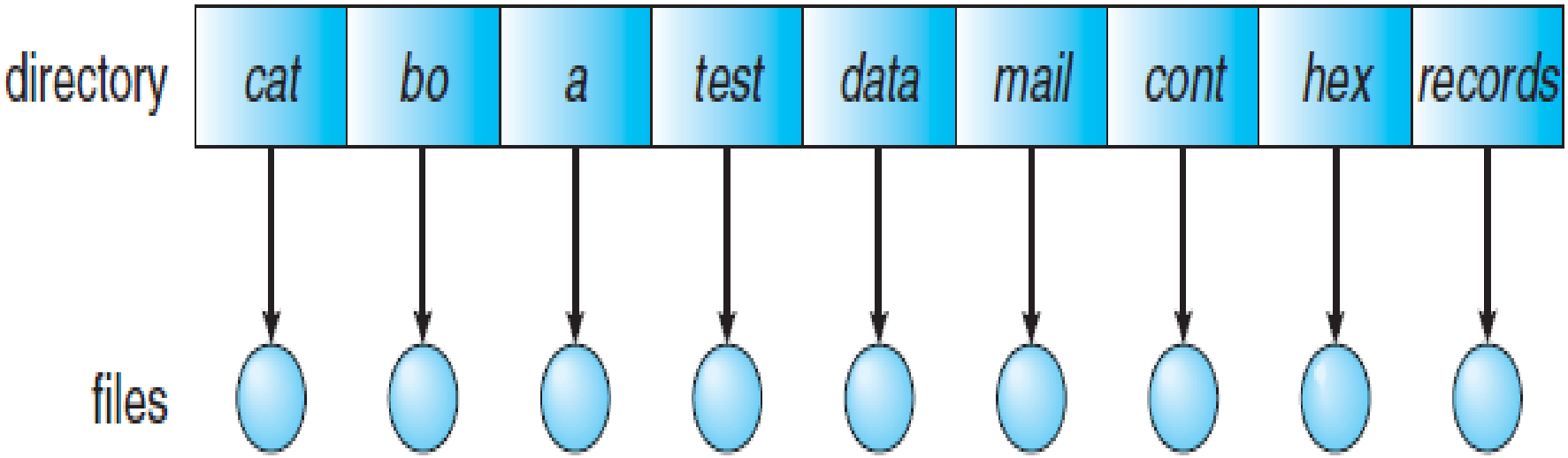
- The most common schemes for defining the logical structure of a directory are as follows:

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

Single-Level Directory

Single-Level Directory

- A *single* directory for all users.

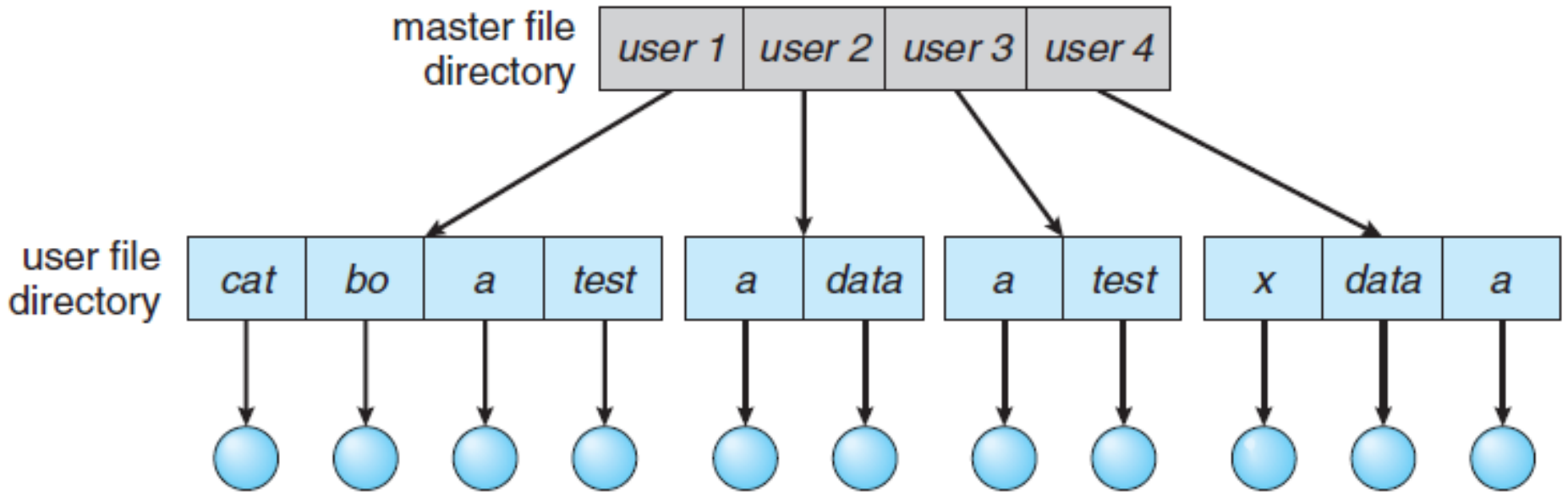


- Characteristics :
  - Naming problem
  - Grouping problem

Two-Level Directory

Two-Level Directory

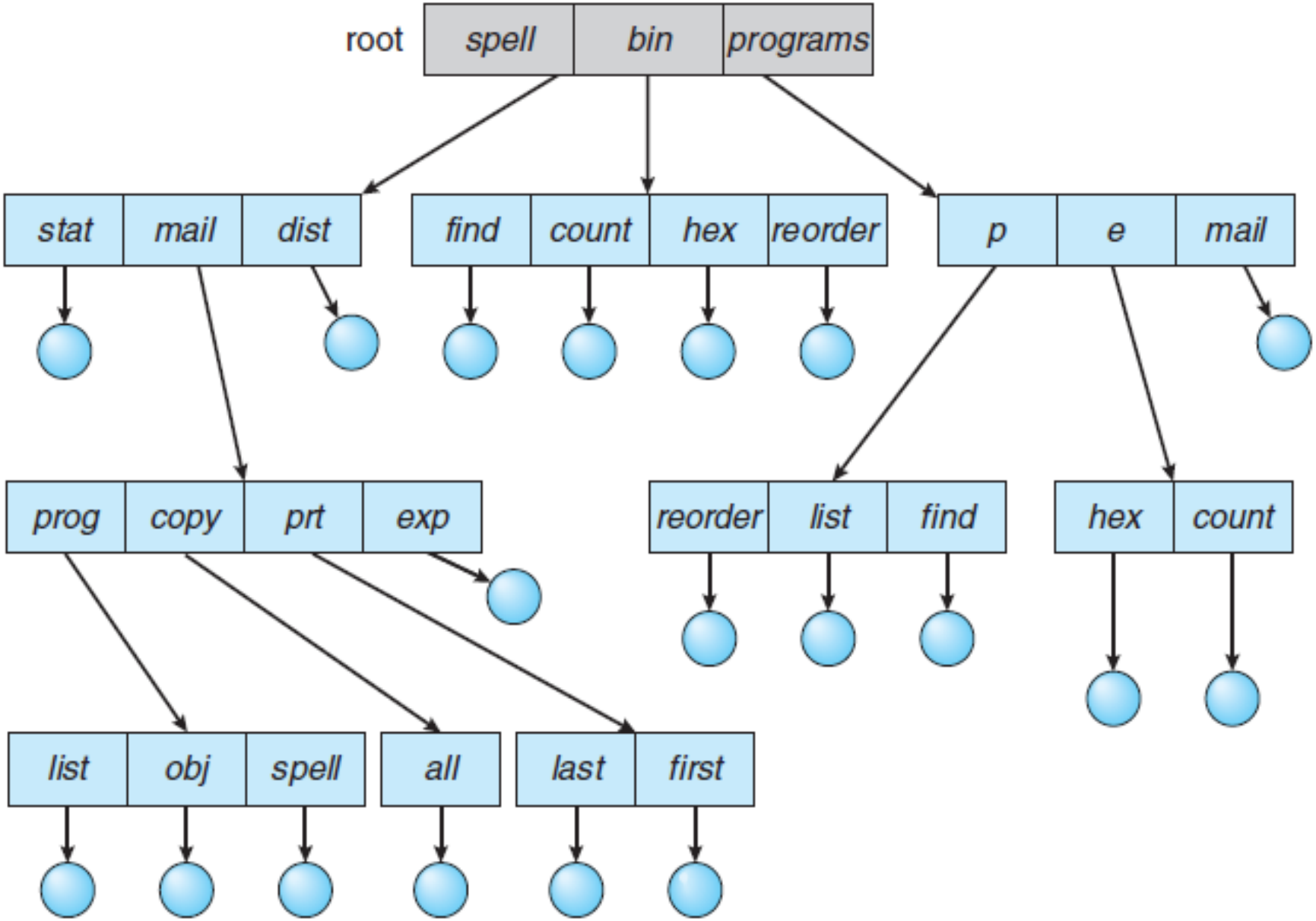
- *Separate* directory for each user.



- Characteristics:
  - Path name
  - Can have the same file name for different user
  - Efficient searching
  - No grouping capability

Tree-Structured Directory

Tree-Structured Directory

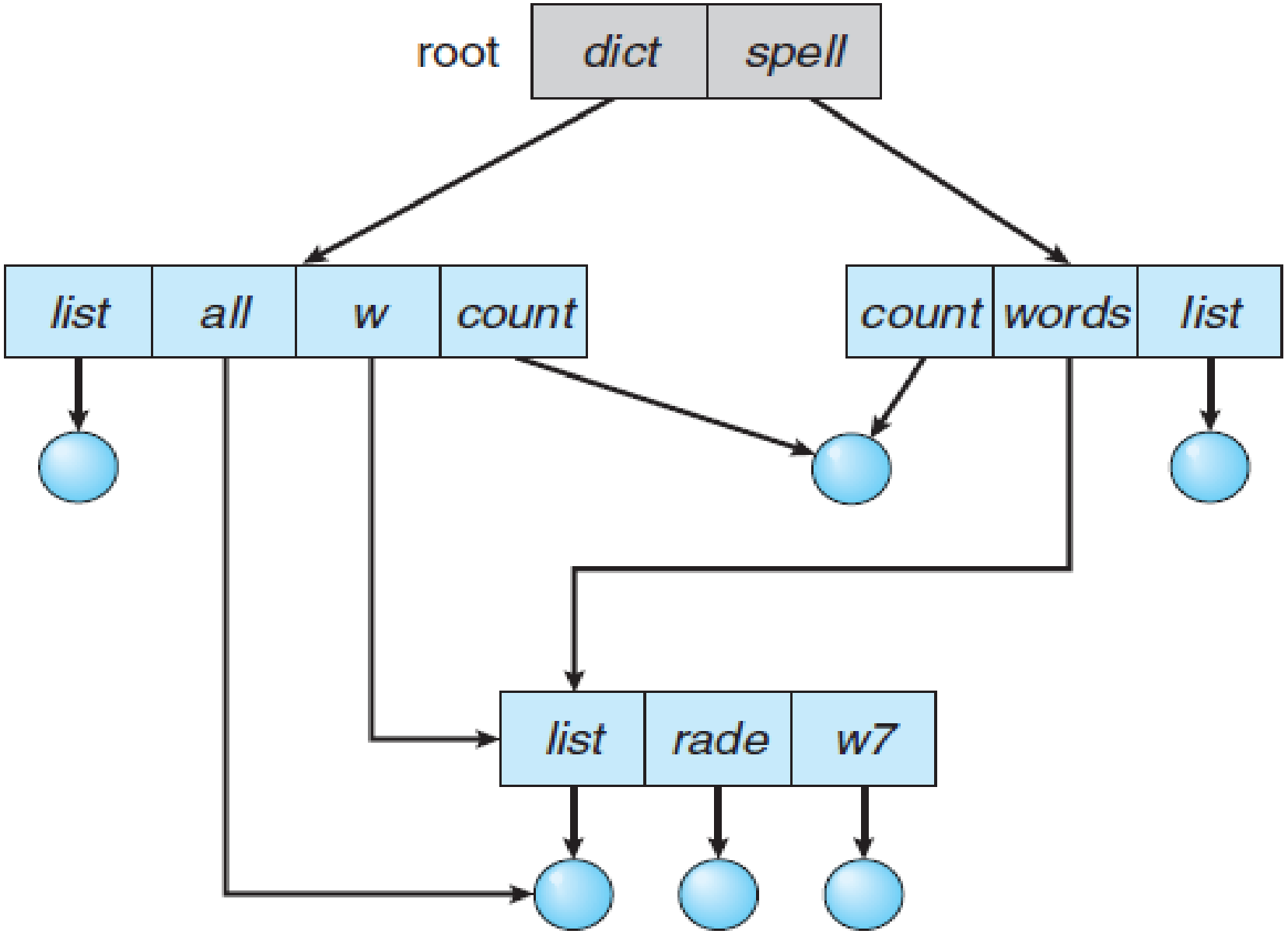


Tree-Structured Directory

- Characteristics:
  - Efficient searching
  - Grouping capability
  - Current directory (working directory)
  - Path names can be of two types: *absolute and relative*.
  - An absolute path name begins at the root and follows a path down to the specified file, giving the directory names on the path.
  - A relative path name defines a path from the current directory.
    - For **example**, if the current directory is root\spel\mail, then the relative path name prt\first refers to the same file as does the absolute path name root\spell\mail\prt\first.

Acyclic-Graph Directory

Acyclic-Graph Directory



Acyclic-Graph Directory

- Characteristics :
  - Links to share files – *only one copy exists*
  - How do we guarantee no cycles?
    - Allow only links to files not subdirectories
    - Garbage collection
    - Every time a new link is added use a cycle detection algorithm to determine whether it is OK.

General Graph Directory

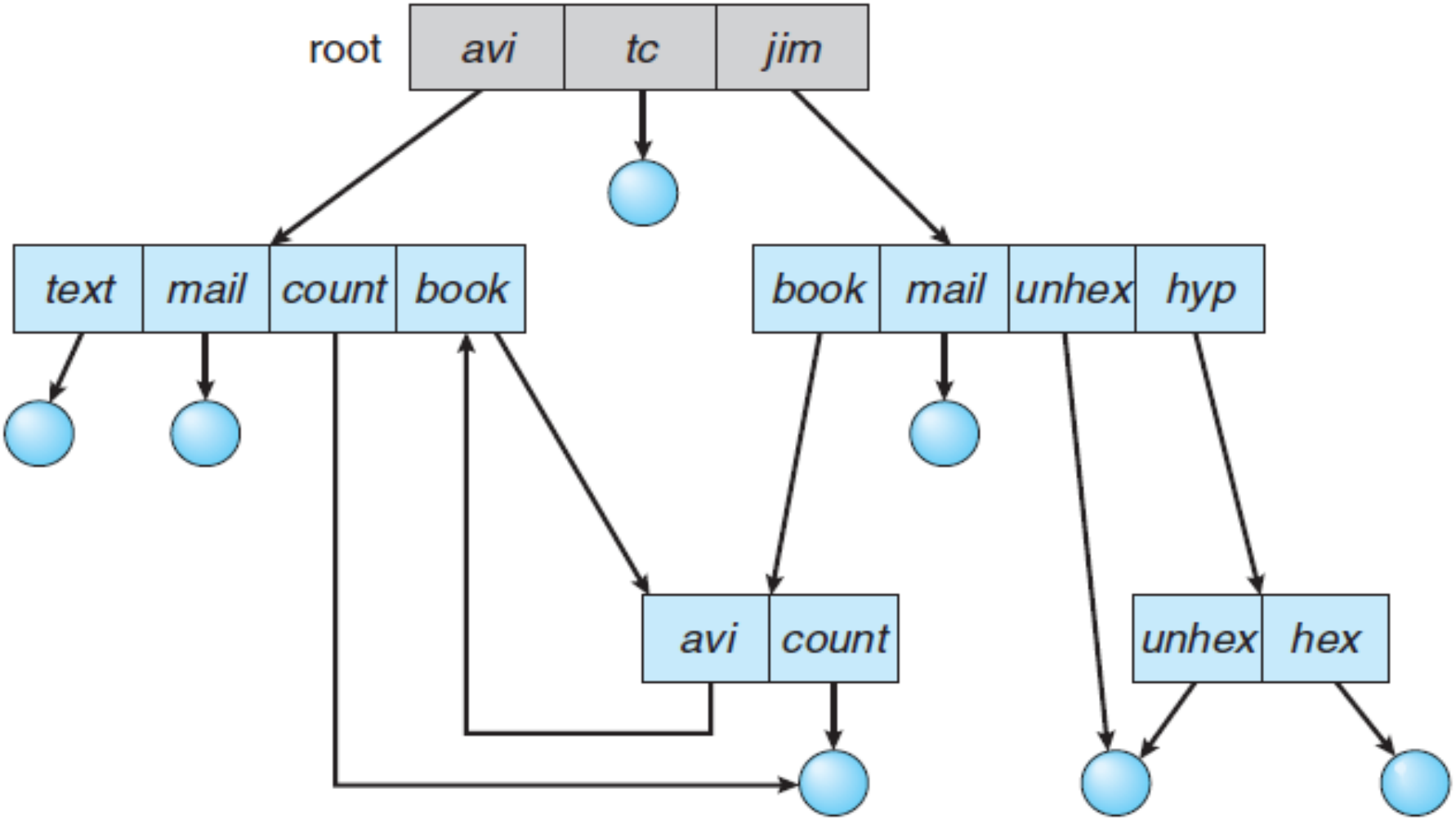
General Graph Directory

- If *cycles are allowed* to exist in the directory, we likewise want to avoid searching any component twice, for reasons of correctness as well as performance.

- A poorly designed algorithm might result in an *infinite loop* continually searching through the cycle and never terminating.

- One solution is to *limit* arbitrarily the number of directories that will be accessed during a search.

General Graph Directory



References

1. Silberschatz, Galvin and Gagne, “Operating Systems Concepts”, Wiley.

2. William Stallings, “Operating Systems: Internals and Design Principles”, 6th Edition, Pearson Education.

3. D M Dhamdhere, “Operating Systems: A Concept based Approach”, 2nd Edition, TMH.

Thank You.