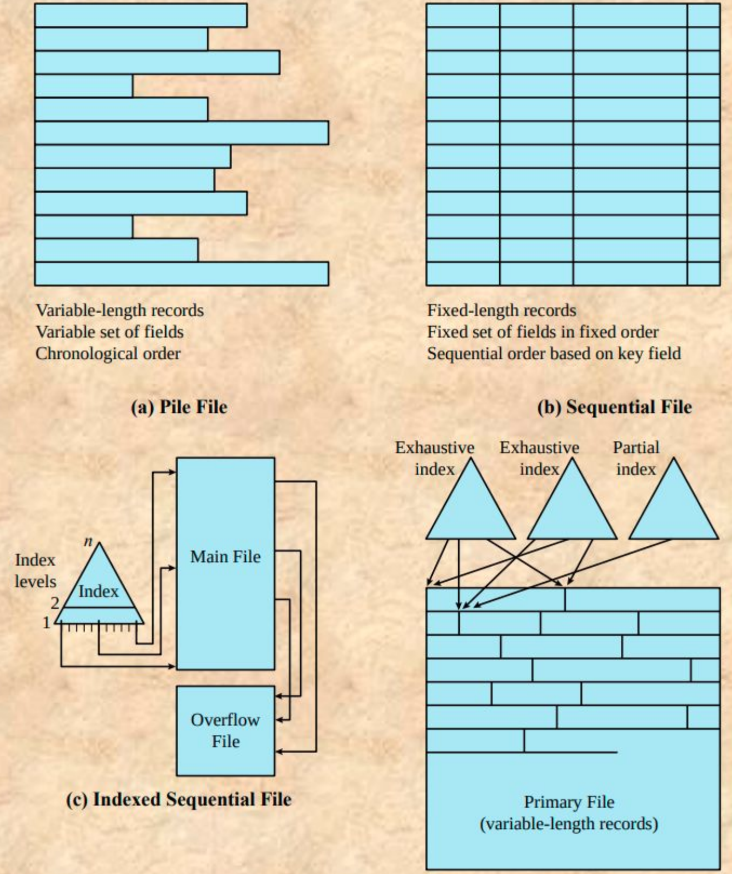
File Organization

DA

File Organization and Access

- ***File organization*** is the logical structuring of the records as determined by the way in which they are accessed
- In choosing a file organization, several criteria are important:
    - short access time
    - ease of update
    - economy of storage
    - simple maintenance
    - reliability

- Priority of criteria depends on the application that will use the file



Variable-length records
Variable set of fields
Chronological order

(a) Pile File

Fixed-length records
Fixed set of fields in fixed order
Sequential order based on key field

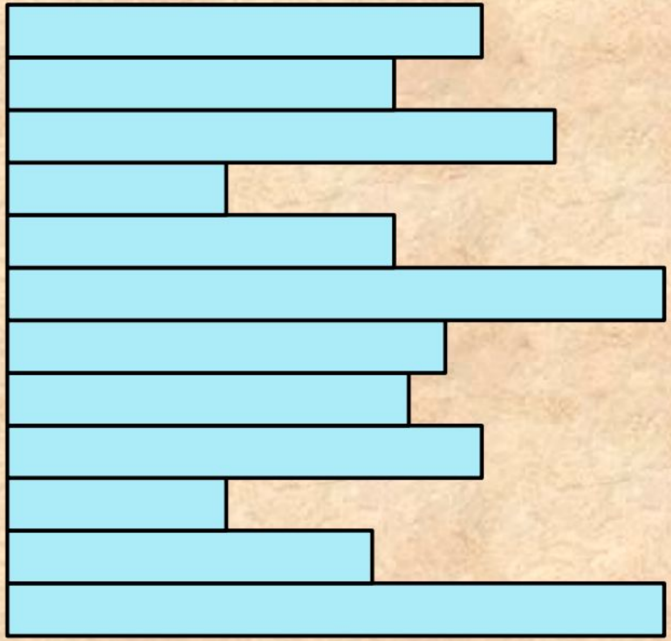
(b) Sequential File

(c) Indexed Sequential File

(d) Indexed File

The Pile

- Least complicated form of file organization
- Data are collected in the order they arrive
- Each record consists of one burst of data
- Purpose is simply to accumulate the mass of data and save it
- Record access is by exhaustive search

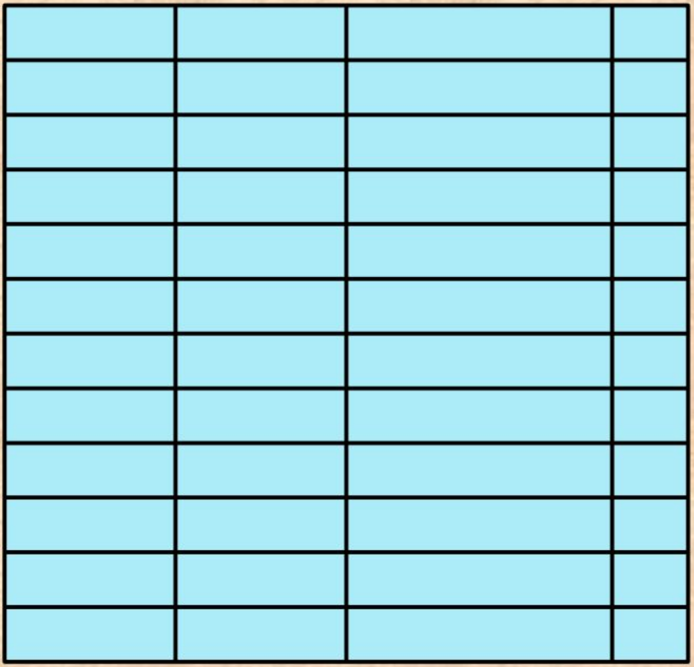


Variable-length records
Variable set of fields
Chronological order

(a) Pile File

The Sequential File

- Most common form of file structure
- A fixed format is used for records
- Key field uniquely identifies the record
- Typically used in batch applications
- Only organization that is easily stored on tape as well as disk

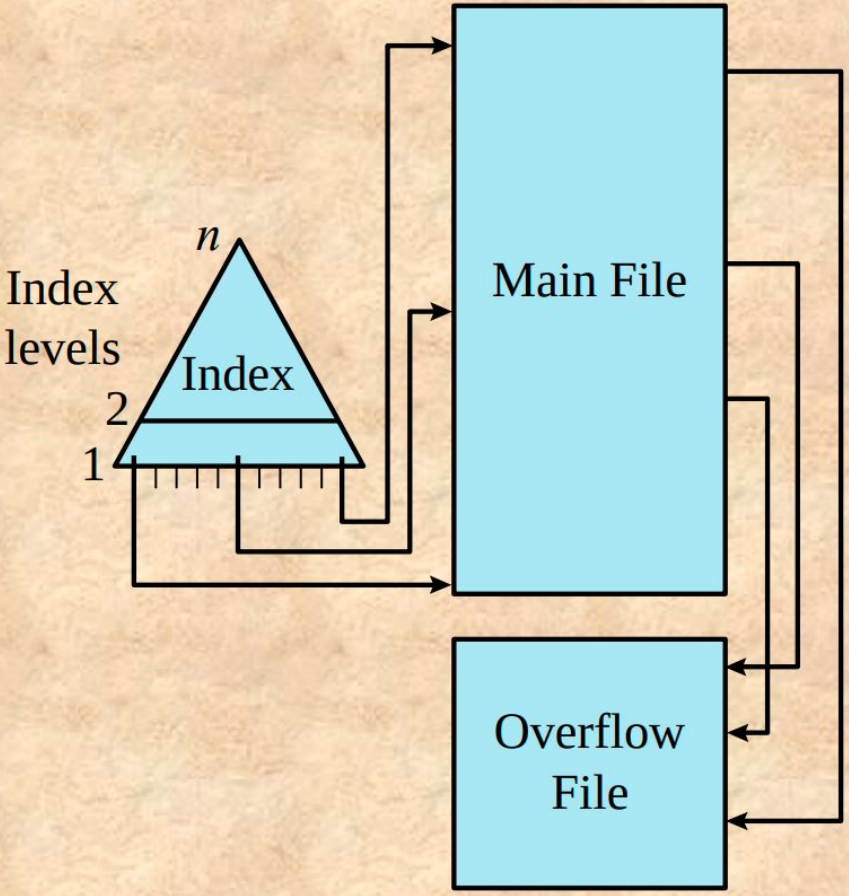


Fixed-length records
Fixed set of fields in fixed order
Sequential order based on key field

(b) Sequential File

Indexed Sequential File

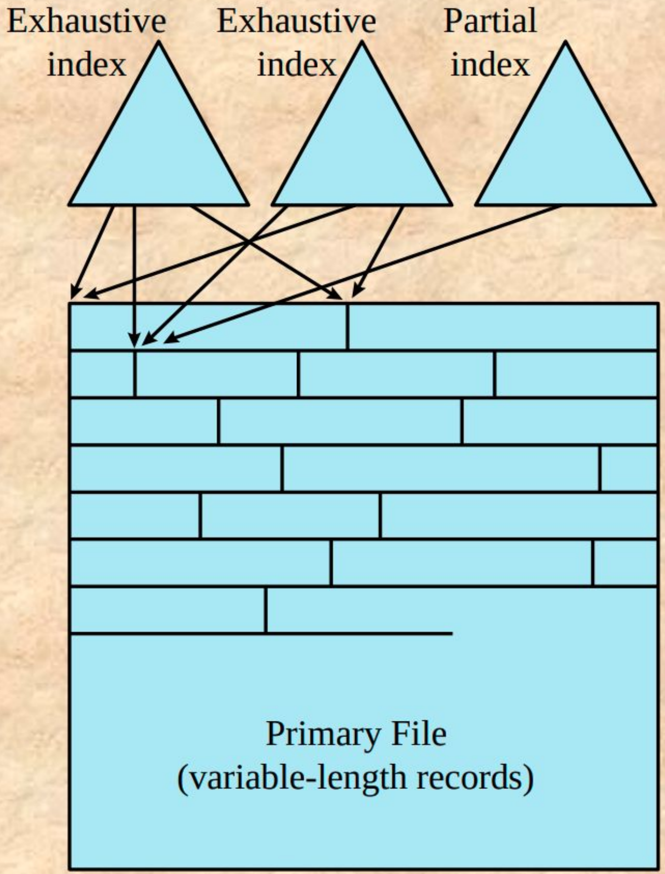
- Adds an index to the file to support random access
- Adds an overflow file
- Greatly reduces the time required to access a single record
- Multiple levels of indexing can be used to provide greater efficiency in access



(c) Indexed Sequential File

Indexed File

- Records are accessed only through their indexes
- Variable-length records can be employed
- Exhaustive index contains one entry for every record in the main file
- Partial index contains entries to records where the field of interest exists
- Used mostly in applications where timeliness of information is critical
- Examples would be airline reservation systems and inventory control systems



(d) Indexed File

Direct or Hashed File

- Access directly any block of a known address
- Makes use of hashing on the key value
- Often used where:
  - very rapid access is required
  - fixed-length records are used
  - records are always accessed one at a time

Examples are:

- directories
- pricing tables
- schedules
- name lists

B-Trees

- A balanced tree structure with all branches of equal length
- Standard method of organizing indexes for databases
- Commonly used in OS file systems
- Provides for efficient searching, adding, and deleting of items

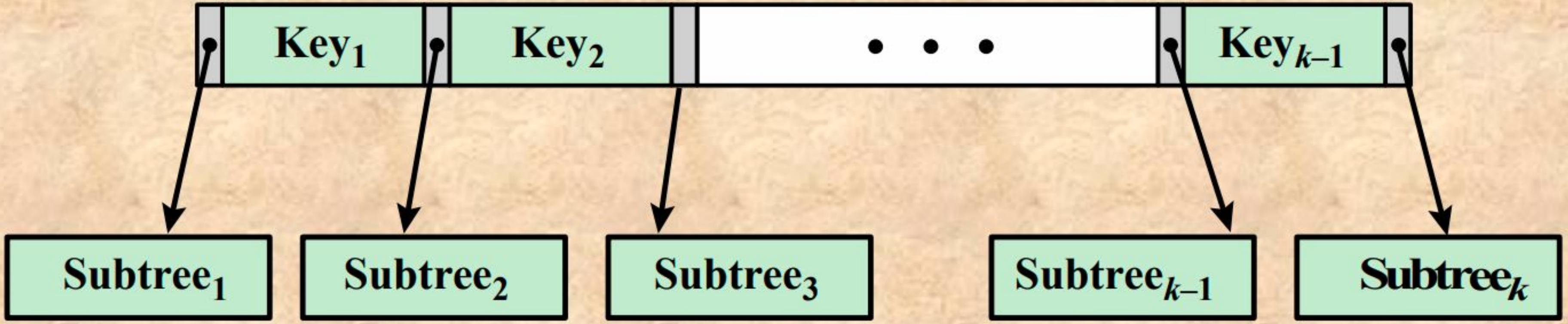
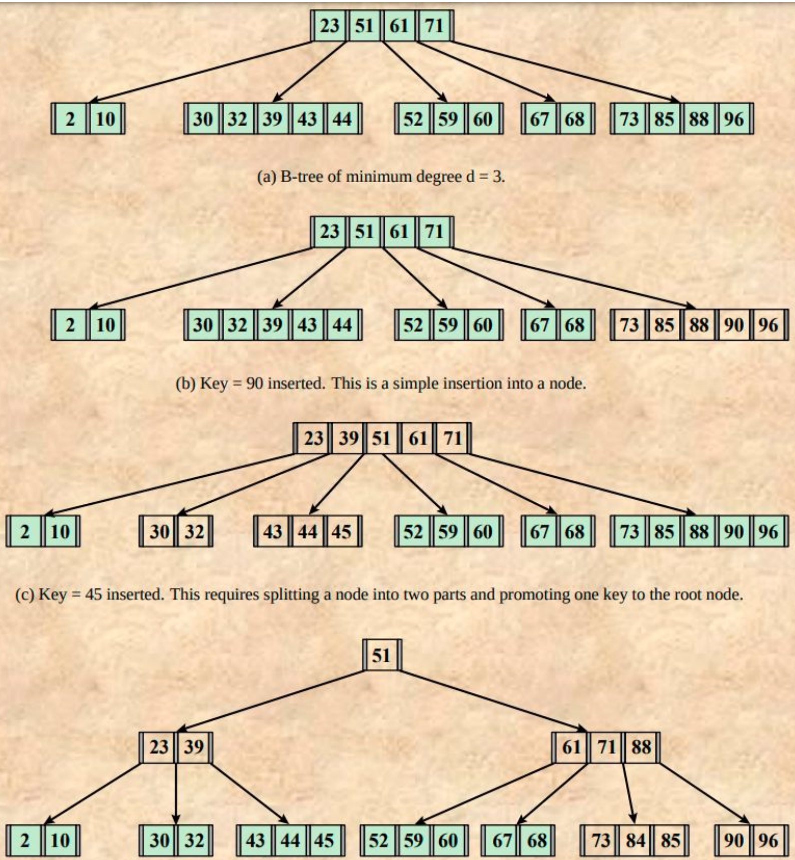


Figure 12.4 A B-tree Node with *k* Children

B-Tree Characteristics

A B-tree is characterized by its minimum degree *d* and satisfies the following properties:

- every node has at most $2d - 1$ keys and $2d$ children or, equivalently, $2d$ pointers
- every node, except for the root, has at least $d - 1$ keys and d pointers, as a result, each internal node, except the root, is at least half full and has at least d children
- the root has at least 1 key and 2 children
- all leaves appear on the same level and contain no information. This is a logical construct to terminate the tree; the actual implementation may differ.
  - a nonleaf node with k pointers contains $k - 1$ keys



(a) B-tree of minimum degree d = 3.

(b) Key = 90 inserted. This is a simple insertion into a node.

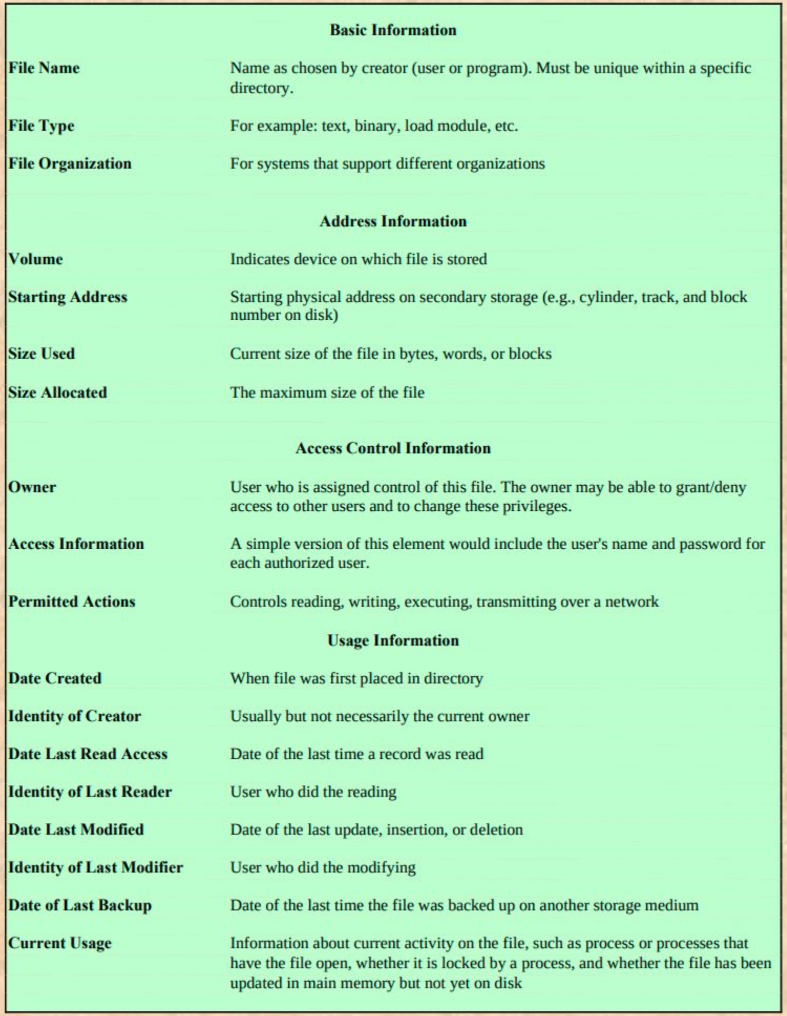
(c) Key = 45 inserted. This requires splitting a node into two parts and promoting one key to the root node.

(d) Key = 84 inserted. This requires splitting a node into two parts and promoting one key to the root node
This then requires the root node to be split and a new root created.

Figure 12.5 Inserting Nodes into a B-tree

Table 12.1

Information Elements of a File Directory

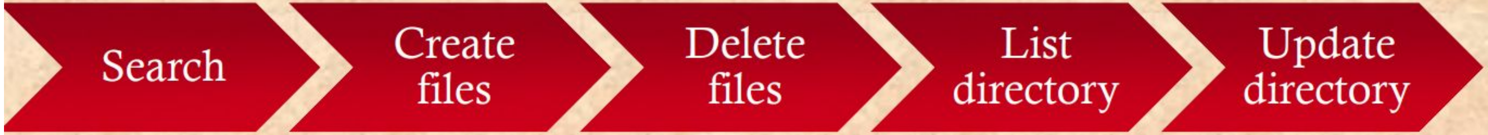


Basic Information	
File Name	Name as chosen by creator (user or program). Must be unique within a specific directory.
File Type	For example: text, binary, load module, etc.
File Organization	For systems that support different organizations
Address Information	
Volume	Indicates device on which file is stored
Starting Address	Starting physical address on secondary storage (e.g., cylinder, track, and block number on disk)
Size Used	Current size of the file in bytes, words, or blocks
Size Allocated	The maximum size of the file
Access Control Information	
Owner	User who is assigned control of this file. The owner may be able to grant/deny access to other users and to change these privileges.
Access Information	A simple version of this element would include the user's name and password for each authorized user.
Permitted Actions	Controls reading, writing, executing, transmitting over a network
Usage Information	
Date Created	When file was first placed in directory
Identity of Creator	Usually but not necessarily the current owner
Date Last Read Access	Date of the last time a record was read
Identity of Last Reader	User who did the reading
Date Last Modified	Date of the last update, insertion, or deletion
Identity of Last Modifier	User who did the modifying
Date of Last Backup	Date of the last time the file was backed up on another storage medium
Current Usage	Information about current activity on the file, such as process or processes that have the file open, whether it is locked by a process, and whether the file has been updated in main memory but not yet on disk

(Table can be found on page 537 in textbook)

Operations Performed on a Directory

- To understand the requirements for a file structure, it is helpful to consider the types of operations that may be performed on the directory:



Search

Create files

Delete files

List directory

Update directory

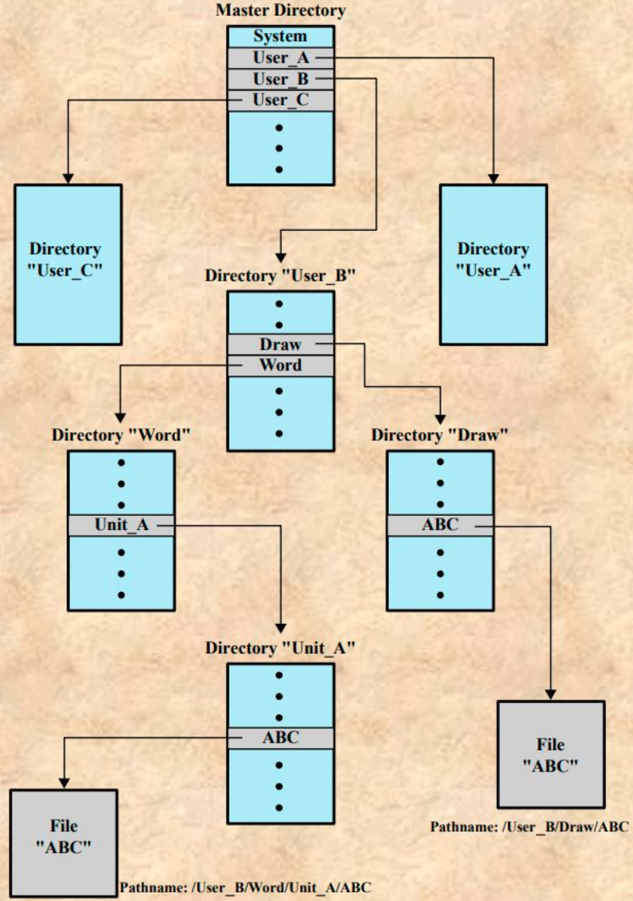


Figure 12.7 Example of Tree-Structured Directory

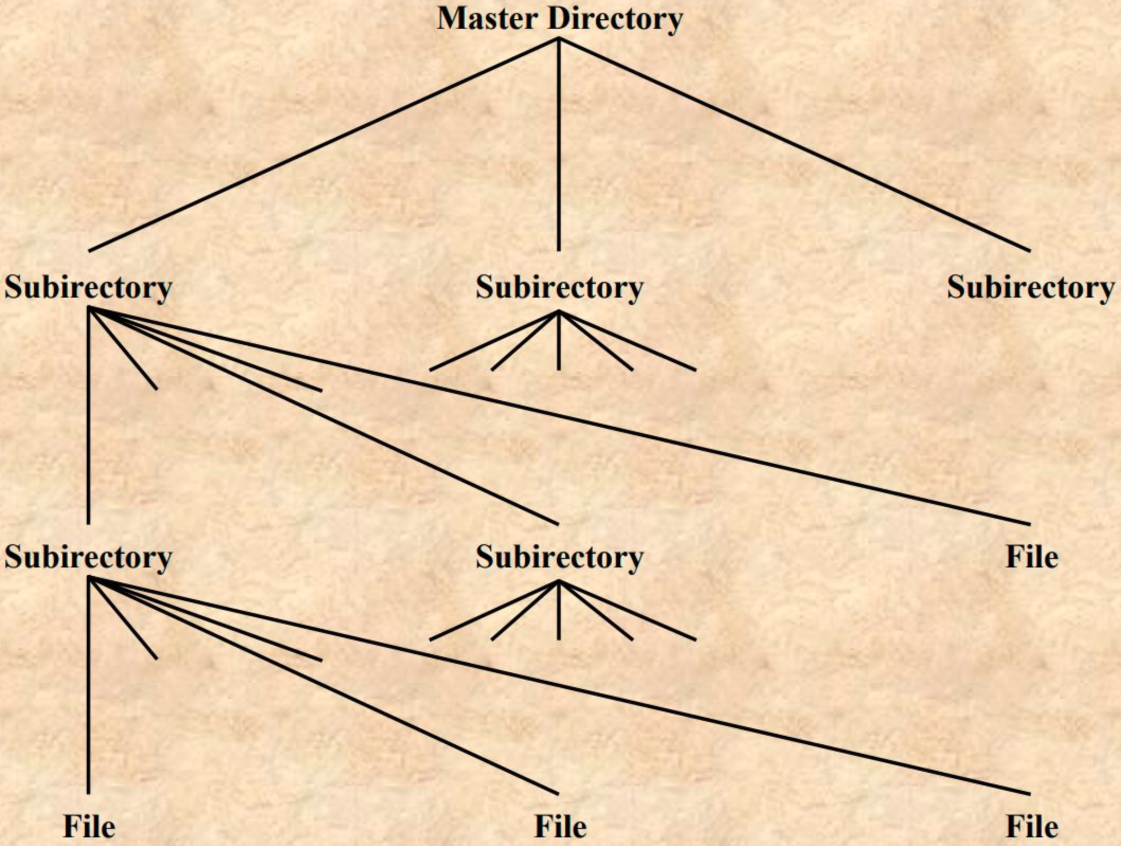
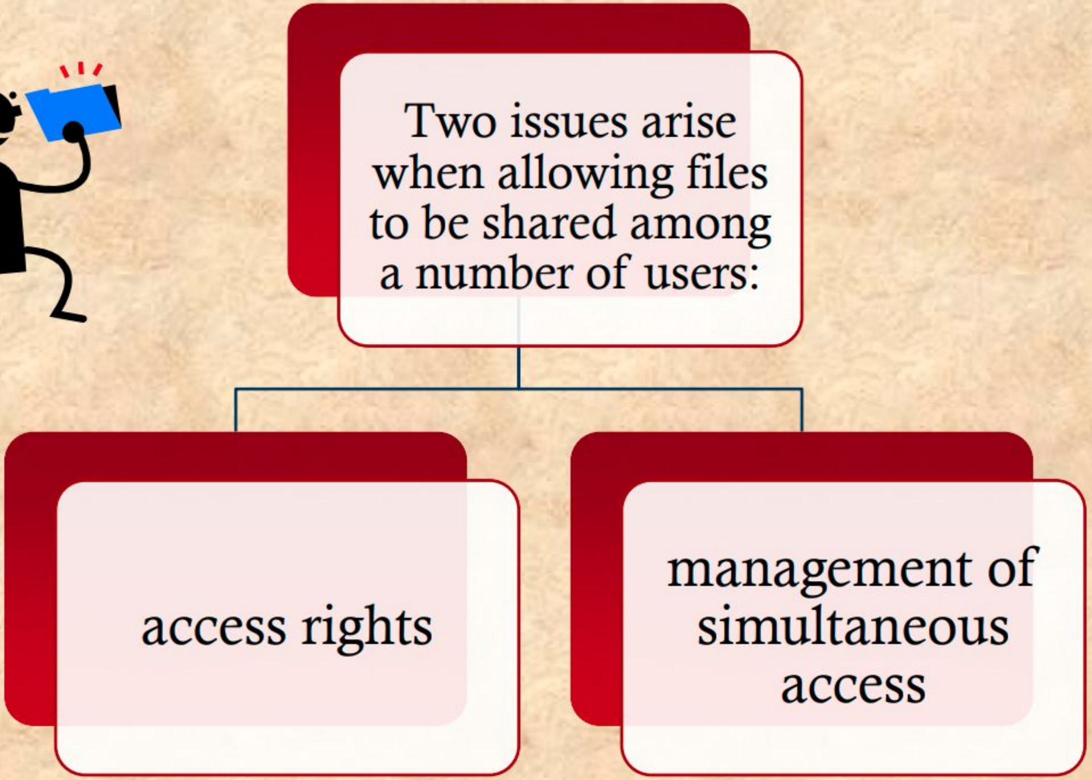


Figure 12.6 Tree-Structured Directory

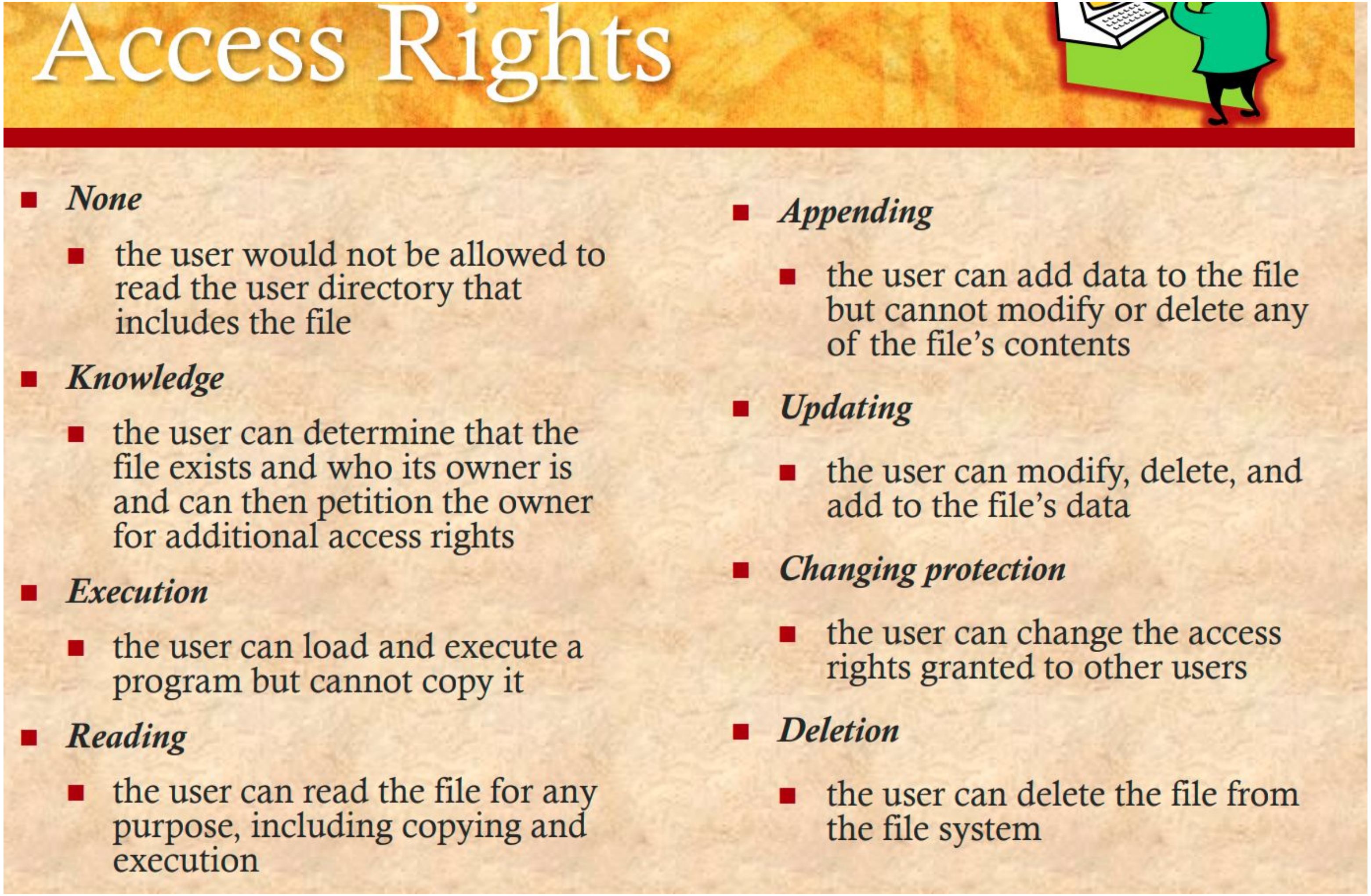
File Sharing



Two issues arise when allowing files to be shared among a number of users:

access rights

management of simultaneous access



Access Rights

- ***None***
  - the user would not be allowed to read the user directory that includes the file
- ***Knowledge***
  - the user can determine that the file exists and who its owner is and can then petition the owner for additional access rights
- ***Execution***
  - the user can load and execute a program but cannot copy it
- ***Reading***
  - the user can read the file for any purpose, including copying and execution
- ***Appending***
  - the user can add data to the file but cannot modify or delete any of the file's contents
- ***Updating***
  - the user can modify, delete, and add to the file's data
- ***Changing protection***
  - the user can change the access rights granted to other users
- ***Deletion***
  - the user can delete the file from the file system

Owner

- usually the initial creator of the file
- has full rights
- may grant rights to others

Specific Users

- individual users who are designated by user ID

User Groups

- a set of users who are not individually defined

All

- all users who have access to this system
- these are public files