# CS420: Operating Systems

# File-System Interface

James Moscola
Department of Physical Sciences
York College of Pennsylvania

# File System

- **OS provides an abstraction of a file on a physical storage device through a file system**

- **File is a named collection of related information that is recorded on a storage device**
  - File Types:
    - Data  (e.g. numeric, character, binary)
    - Program (e.g. an executable)

# File Structure

- **Files may have a variety of different structures**
  - None - simply a sequence of words, bytes
  - Simple record structure consisting of lines of information
  - Complex structures like formatted documents

- **Either the operating system or the program that generates the file determines the structure of the file**

# File Attributes

- **Name – the symbolic file name; the only information kept in human-readable form**
- **Identifier – unique tag (number) that identifies file within file system**
- **Type – needed for systems that support different types**
- **Location – pointer to file location on a device**
- **Size – current file size in bytes (or blocks)**
- **Protection – access-control information; controls who can do reading, writing, executing**
- **Time, date, and user identification – Info may be maintained for creation time, last modification time, and last time accessed; data for protection, security, and usage monitoring**

# Directory Structure

- **Information about files is kept in the directory structure**
  - An entry exists for each file that typically consists of:
    - the filename
    - the file’s unique identifier
  - The file identifier is used to locate other attributes of a file

- **Directory structure is stored on disk**

# File Operations

- **File is an abstract data type that allows for the following operations:**
  - Create
  - Write
  - Read
  - Reposition within file
  - Delete
  - Truncate

- **open($F_i$) – search the directory structure on disk for entry $F_i$, and move the content of entry to memory**

- **close($F_i$) – move the content of entry $F_i$ in memory to directory structure on disk**

# Open Files

- **An open-file table is a system-wide table that contains an entry for each file that is open on the system**

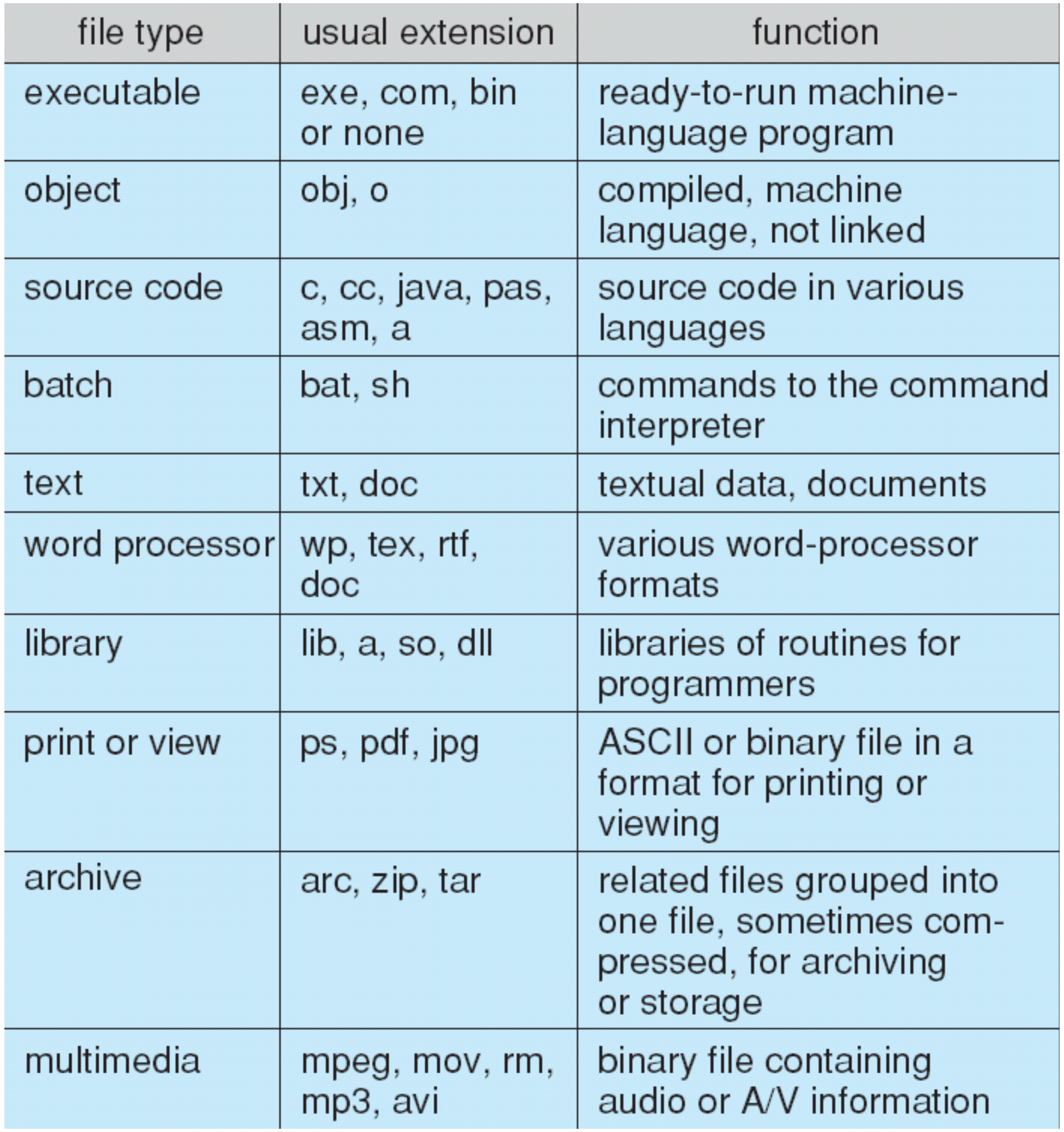
- **Several pieces of data are needed to manage open files:**
  - File pointer - pointer to last read/write location, per process that has the file open
  - File-open count - counter of number of times a file is open – to allow removal of data from open-file table when last processes closes it
  - Disk location of the file - cache of data access information
  - Access rights - per-process access mode information

# Open File Locking

- **Provided by some operating systems and file systems**
- **Mediates access to a file**
- **Mandatory or advisory:**
  - Mandatory – access is denied depending on locks held and requested
  - Advisory – processes can find status of locks and decide what to do

# File Types – Name, Extension

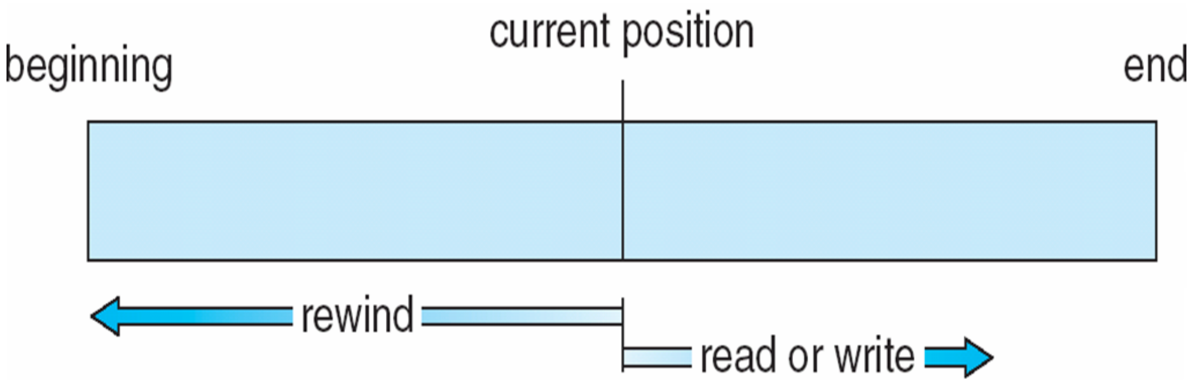
- **File names can contain extensions**
  - Some operating systems use the file type to determine how to handle the file
  - Some operating systems don't actually use the extension, they are merely there to provide hints to the user as to what the type of the file is



| 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 compressed, for archiving or storage |
| multimedia | mpeg, mov, rm, mp3, avi | binary file containing audio or A/V information |

# Access Methods

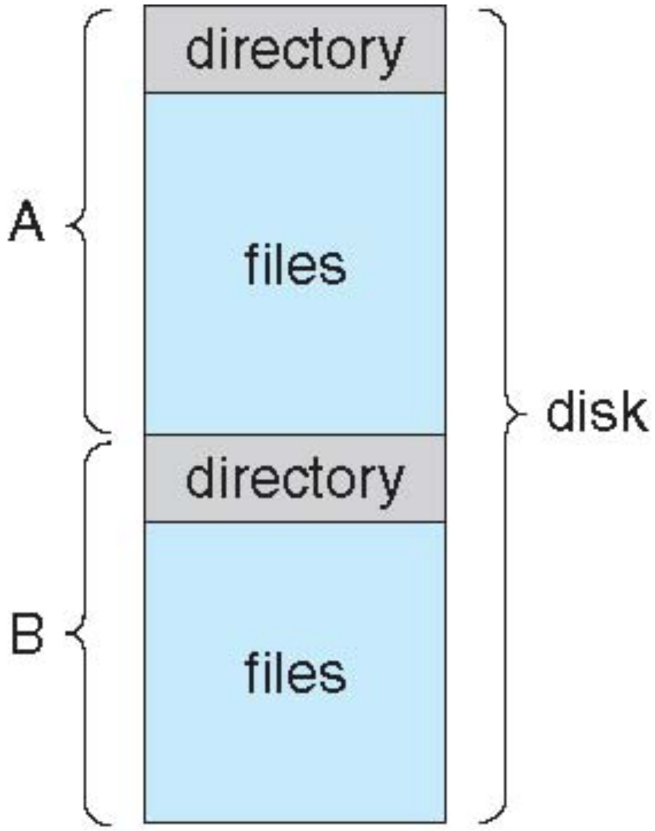
- **Information stored in files can be accessed in different ways:**
  - Sequential Access - information in the file is processed in order, one record after another (size of record is dependent on OS, typically a byte)



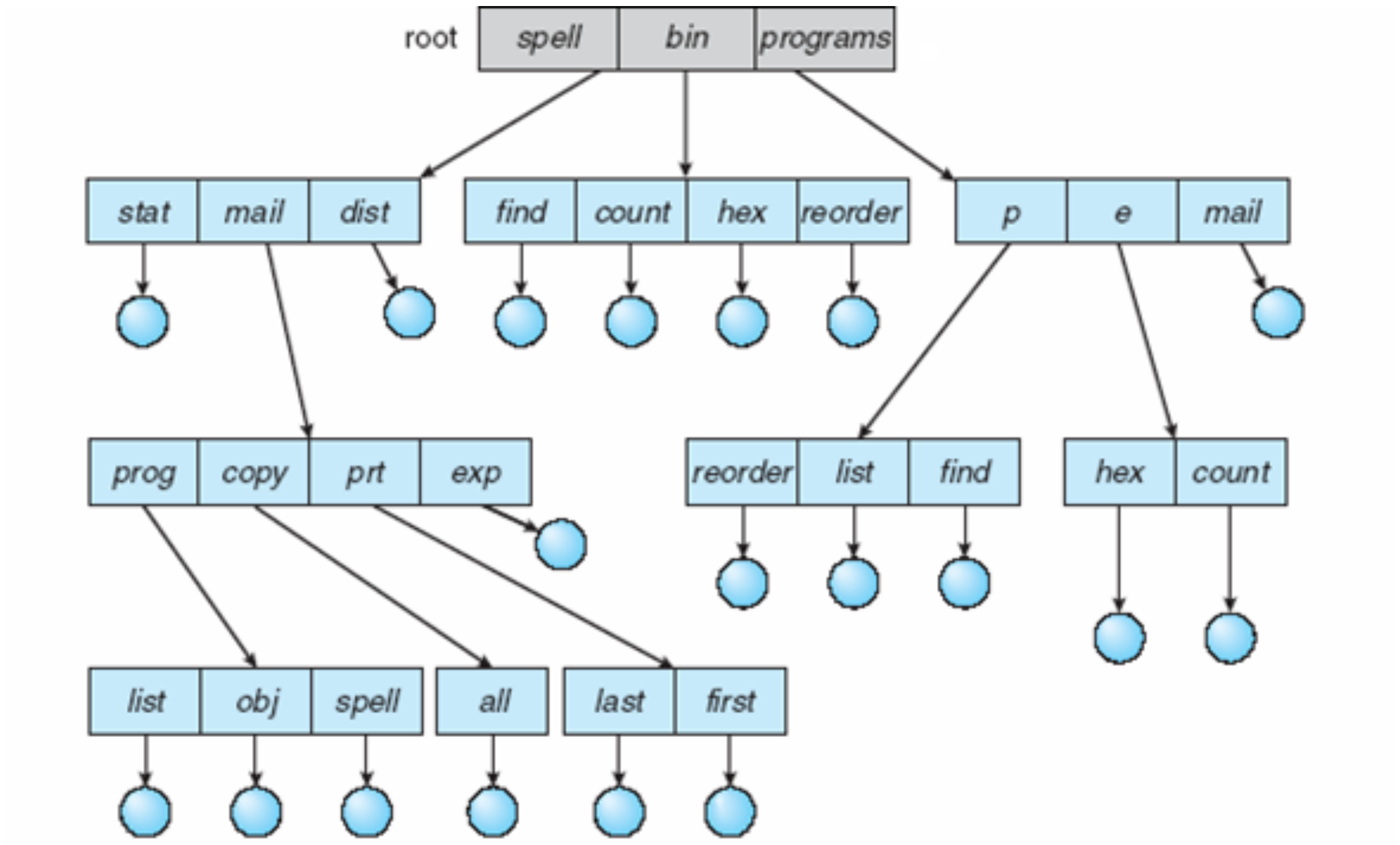
  - Direct Access - records in a file can be accessed in any order (very useful for databases)

# Directory

- **Directory is a special type of 'file'**
  - Essentially a symbol table that translates file names into an entry on disk
- **Both the files and the directory structure reside on disk**
- **The following are operations that can be performed on a directory**
  - Search for a file
  - Create a file
  - Delete a file
  - List a directory
  - Rename a file
  - Traverse the file system



# Tree-Structured Directories



# File System Mounting

- **A file system must be mounted before it can be accessed**
  - Provide the operating system with the name of the device and a mount point
    - Mounting point is typically an empty directory on local machine

- **The operating system will verify that the device contains a valid file system before mounting**
  - On some systems, you must specify the file system on the device being mounted

# File Sharing

- **Sharing of files on multi-user systems is desirable**
  - User IDs - identify users, allowing permissions and protections to be per-user
  - Group IDs - allow users to be in groups, permitting group access rights
- **Sharing may be done through a protection scheme**
- **On distributed systems, files may be shared across a network**

# 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
- **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 (Network File System) is standard UNIX client-server file sharing protocol
  - CIFS (Common Internet File System) is standard Windows protocol
  - Standard operating system file calls are translated into remote calls

# Protection

- **File owner/creator should be able to control access to file**
  - What can be done to the file?
  - Who can do what to the file?
- **Types of access that are typically controlled**
  - Read
  - Write
  - Execute
  - Append
  - Delete
  - List

# Access Lists and Groups

- **Access Lists can be used to control mode of access to a file (i.e. read, write, execute permissions)**

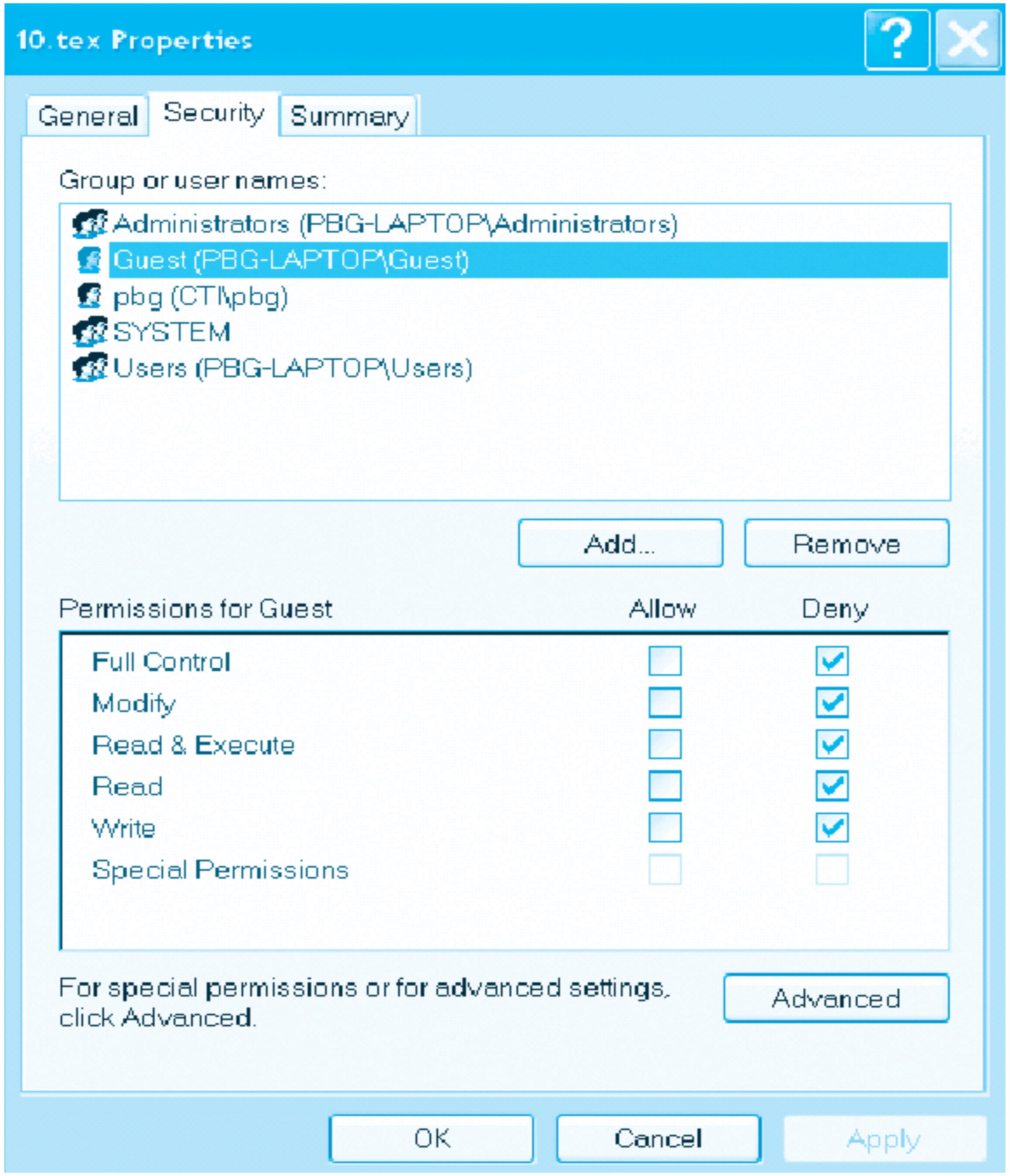
- **Three classes of users (use 3 bits per class)**

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

- **Users can be added to a group that allows them access to certain files**

- **On Unix/Linux systems, mode of access can be changed by owner using `chmod` command**

# Windows XP Access-Control List Management



10.tex Properties

General | Security | Summary

Group or user names:

- Administrators (PBG-LAPTOP\Administrators)
- Guest (PBG-LAPTOP\Guest)
- pbg (CTI\pbg)
- SYSTEM
- Users (PBG-LAPTOP\Users)

| Permissions for Guest | Allow | Deny |
|---|---|---|
| Full Control | □ | ☑ |
| Modify | □ | ☑ |
| Read & Execute | □ | ☑ |
| Read | □ | ☑ |
| Write | □ | ☑ |
| Special Permissions | □ | □ |

For special permissions or for advanced settings, click Advanced.