OPERATING SYSTEM (315319)

UNIT V

FILE MANAGEMENT

Unit Scope

- File concepts: attributes, operations, file types and file-system structure
- File access methods: sequential and direct
- File allocation methods: contiguous, linked and indexed
- Directory structures: single-level, two-level and tree-structured
- Basic file/directory permissions needed to support TLO 5.3

Learning Outcomes

TLO	Expected capability
5.1	Explain the structure of a given file system with an example.
5.2	Describe the mechanism of file access methods.
5.3	Explain the procedure to create access directories and assign file access permissions.

Assessment weightage Unit V carries 10 marks and 8 learning hours. The specification table gives R = 2, U = 4 and A = 4.

1. File Concepts

- A file is a named collection of related information stored on secondary storage.
- Examples include a text document, image, program source file, executable program or database file.
- The operating system provides a uniform interface so users and programs can create, access, modify and delete files.

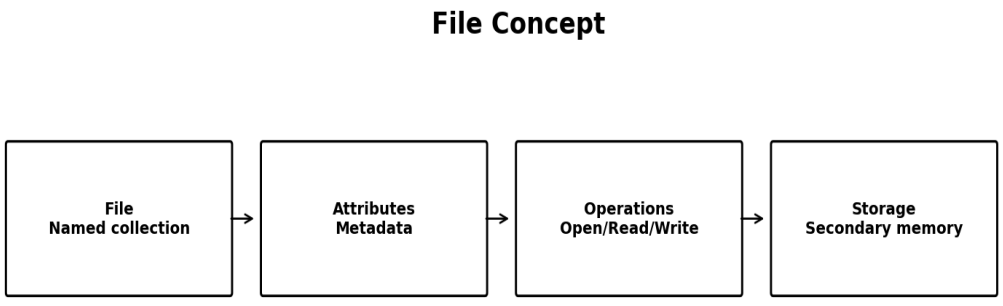

Fig. 5.1 File concept: data + metadata + operations

1.1 File Attributes

Attribute	Meaning / example
Name	Human-readable file name, such as report.docx.
Identifier	Unique internal identifier used by the file system.
Type	Indicates the kind or format of the file; often associated with extension.
Location	Information identifying where the file's data blocks are stored.
Size	Current file size, usually in bytes.
Protection / permissions	Controls who can read, write or execute/search the file.
Time and ownership	Creation/modification/access information and owner/group information.

Exam Focus File attributes describe a file; file operations describe what the OS can do with the file.

1.2 Common File Operations

Operation	Purpose
Create	Make a new file and its file-system entry.
Open	Prepare a file for access and return a handle/file descriptor.
Read	Copy data from file to memory.
Write	Store data from memory into the file.
Append	Add data at the end of a file.
Seek / reposition	Move the current file position.
Close	Release the open-file resources.
Rename	Change the file name.
Delete	Remove the file and release its allocated storage.

Exam Focus The Winter 2025 paper asks for any four file operations; the Summer 2026 paper asks for any four file operations.

1.3 File Types and Extensions

File type	Typical extension	Example / use
Text	.txt	Plain text
Source code	.c, .cpp, .java, .py	Program source
Document	.docx, .pdf	Office document / PDF
Image	.jpg, .png	Picture
Audio	.mp3, .wav	Sound
Video	.mp4, .mkv	Video
Executable	.exe (Windows), no fixed extension requirement on Linux	Program

- An extension is a naming convention that helps users and applications identify a file's format. It is not, by itself, a guarantee of the file's actual contents.

Exam Focus Winter 2025 asks for file types with usual extensions and the need for a file extension.

2. File-System Structure

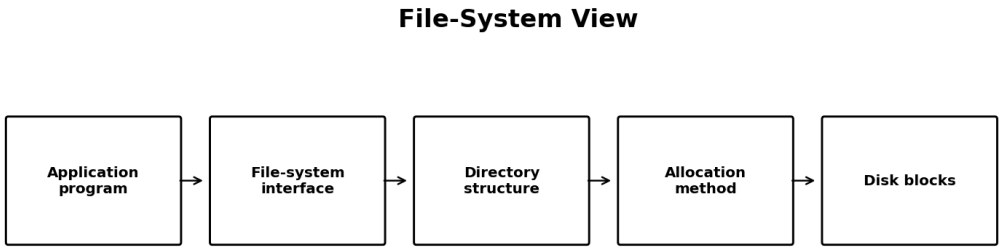

Fig. 5.2 Simplified file-system view

- The file system organizes files and directories on secondary storage.
- A directory maps names to file-system objects and helps users locate files.
- The OS uses file-system metadata and allocation information to find the data blocks belonging to a file.
- The file-system interface hides low-level disk-block details from application programs.

3. File Access Methods

3.1 Sequential Access

- Data is processed in order, from the current position toward the end.
- The next read/write normally operates on the next portion of the file.
- It is suitable for logs, text streams, audio/video streams and sequential data processing.

3.2 Direct (Random) Access

- The file is viewed as a set of numbered or addressable blocks/records.
- A program can move directly to the required position instead of reading all previous data.
- It is useful for databases, large binary files and applications that repeatedly access different records.

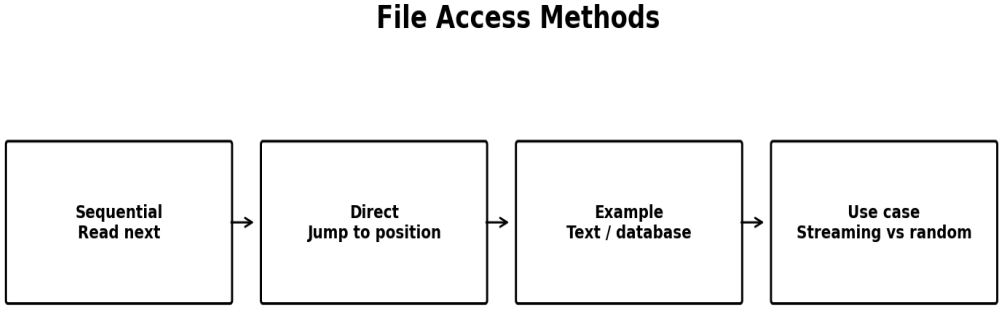

Fig. 5.3 Sequential vs direct access

Feature	Sequential	Direct
Order	Usually in order	Any required position
Speed for random record	Poorer	Better
Typical use	Logs, streams	Databases, indexed data
Operation	Read next / advance	Seek to position and read/write
Exam Focus Summer 2026 specifically asks: “Describe direct access method with one example.”		

4. File Allocation Methods

- File allocation determines how a file's data blocks are placed on secondary storage.
- The syllabus requires three methods: contiguous, linked and indexed allocation.

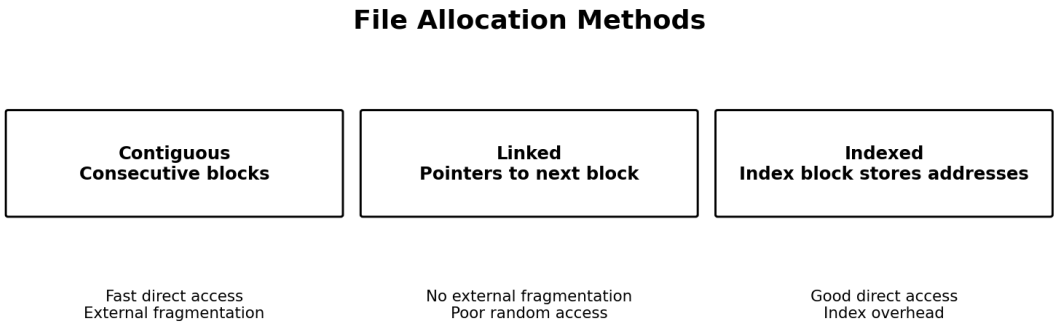

Fig. 5.4 Allocation methods at a glance

4.1 Contiguous Allocation

- All blocks of a file are stored in consecutive disk locations.
- The directory entry can store the starting block and the number of blocks.
- For a file beginning at block 10 and using four blocks, the blocks are 10, 11, 12 and 13.

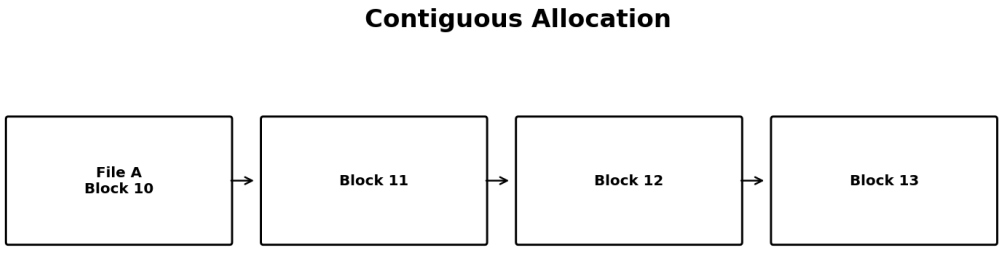

Fig. 5.5 Contiguous allocation

Advantages	Disadvantages
Simple management	External fragmentation
Very good sequential access	File growth can be difficult
Very good direct access	May need relocation or reallocation for growth
Low pointer overhead	Free contiguous space may be hard to find
Exam Focus Summer 2026 asks for disadvantages of contiguous file allocation.	

4.2 Linked Allocation

- A file is stored as a linked chain of blocks that can be located anywhere on the disk.
- Each data block contains a pointer to the next block, with the final pointer marking the end.
- A directory entry generally stores the first block; the chain is followed to reach later blocks.

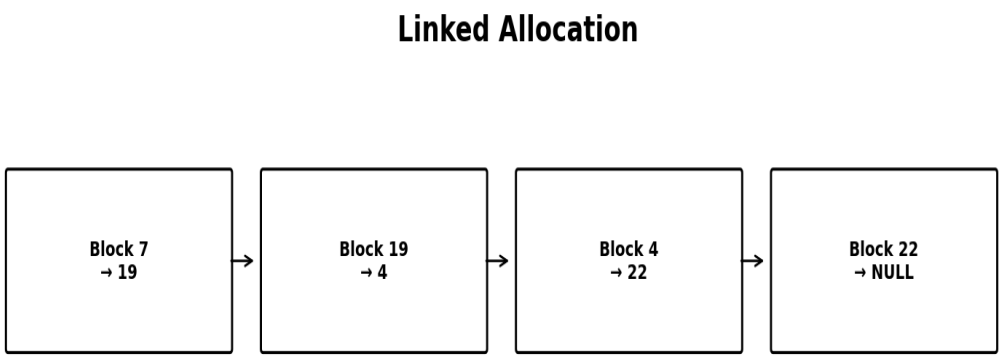

Fig. 5.6 Linked allocation

Advantages	Disadvantages
No external fragmentation requirement for consecutive blocks	Poor direct/random access
File can grow easily if free blocks exist	Pointer storage overhead
Blocks may be anywhere on disk	A damaged pointer can break the chain

4.3 Indexed Allocation

- An index block contains the addresses of the data blocks belonging to the file.
- The directory entry points to the index block.
- Data blocks can be located anywhere on disk, while the index provides the mapping needed to find them.

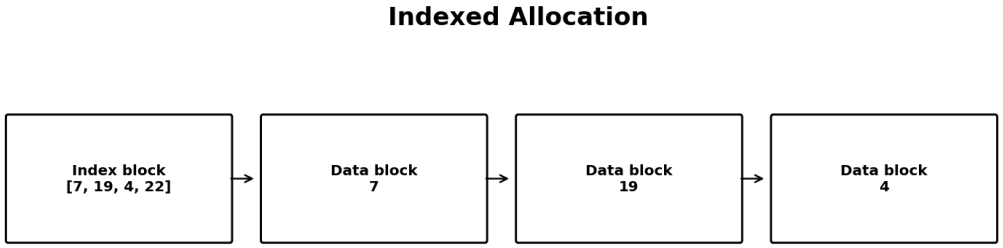

Fig. 5.7 Indexed allocation

Advantages	Disadvantages
Supports direct/random access	Index-block overhead
No need for contiguous data blocks	Very large files may need multi-level indexing
File can grow using free blocks	Extra access may be needed to read the index

4.4 Allocation Comparison

Feature	Contiguous	Linked	Indexed
Block placement	Consecutive	Anywhere	Anywhere
Sequential access	Excellent	Good	Good
Direct access	Excellent	Poor	Good
External fragmentation	Possible	No consecutive-space requirement	No consecutive-space requirement
Growth	Difficult	Easy	Easy
Extra pointer/index overhead	Low	Pointer in each block	Index block(s)

Exam Focus Winter 2025 asks to differentiate linked and indexed allocation. Prepare at least four comparison points.

5. Directory Structure

- A directory is a special file-system structure that stores names and information needed to locate files and/or subdirectories.
- Directory organization determines how easily users can group, locate and manage files.

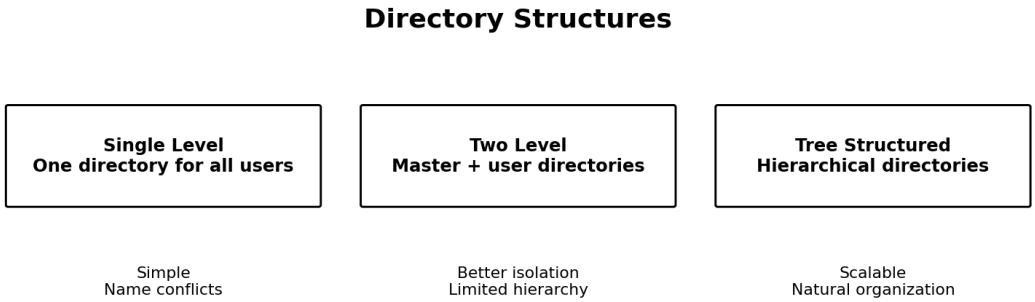

Fig. 5.8 Directory structures

5.1 Single-Level Directory

- All files are kept in one directory.
- It is simple and easy to implement.
- It becomes difficult when many users/files exist because name conflicts can occur and grouping is limited.

Advantages	Disadvantages
Simple	Name conflicts
Easy to understand	No user-wise grouping
Small overhead	Poor scalability

5.2 Two-Level Directory

- A master directory identifies users; each user has a separate user-file directory.
- Two users can have files with the same name because their directories are separate.
- It improves isolation compared with a single-level directory, but sharing files between users is less straightforward.

Advantages	Disadvantages
Avoids cross-user name conflicts	Limited hierarchy
User-wise organization	File sharing can be less convenient
Better isolation	Not as flexible as a tree
Exam Focus Winter 2025 asks for the two-level directory structure with advantages and disadvantages.	

5.3 Tree-Structured Directory

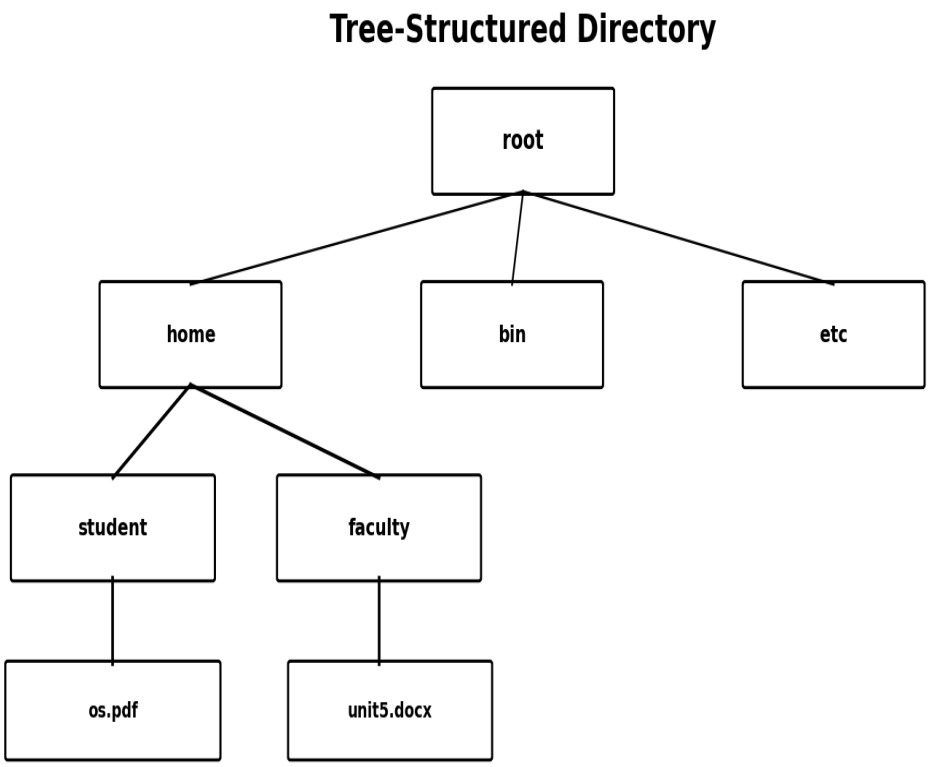

Fig. 5.9 Tree-structured directory

- Directories can contain files and other directories (subdirectories).
- The root directory is the starting point of the hierarchy.
- A path identifies an object by describing its location in the tree.
- Absolute path starts from the root; relative path starts from the current directory.

Advantages	Disadvantages
Natural hierarchical organization	More complex than single-level
Scales to many files/users	Requires path management
Easy grouping by project/topic	Permissions and sharing need careful management
Exam Focus Summer 2026 asks for the tree-structured directory with a diagram.	

6. File and Directory Access Permissions

TLO Support The syllabus TLO 5.3 explicitly expects students to explain how to create access directories and assign file access permissions. Permissions are therefore included as a supporting section even though the theory subtopic list focuses on file concepts, allocation and directories.

6.1 Basic Linux Permission Model

- Permissions are commonly specified for three classes: owner (user), group and others.
- The basic permissions are read (r), write (w) and execute (x).
- For a file, execute means permission to run it when it is an executable program; for a directory, execute/search permission allows traversal through the directory.

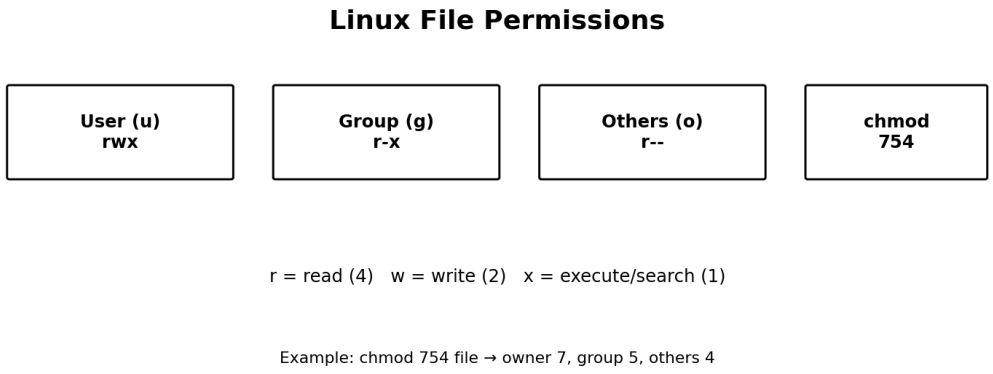

Fig. 5.10 Linux file permissions

Symbol	Numeric value	Meaning
r	4	Read
w	2	Write
x	1	Execute / directory search

6.2 Numeric Permission Example

- 7 = 4+2+1 = read + write + execute.
- 5 = 4+0+1 = read + execute.
- 4 = 4+0+0 = read only.
- Therefore, 754 means owner=rwx, group=r-x, others=r--.

6.3 Creating Directories and Assigning Permissions

Command	Purpose
mkdir project	Create a directory named project.
mkdir -p project/unit5	Create a directory and required parent directories.
ls -l	Display file/directory details including permissions.
chmod 754 report.txt	Assign numeric permissions.
chmod u+rwx,g+rx,o+r report.txt	Assign permissions symbolically.
cd project	Enter a directory.
pwd	Display the current directory path.

Practical note Permission changes should be demonstrated only on files/directories owned by the student or where the student has authorized access.

7. File-System Path Concepts

Term	Meaning	Example
Root	Top of a file-system hierarchy.	/
Absolute path	Complete path starting at root.	/home/student/unit5
Relative path	Path interpreted from current directory.	unit5/notes.txt
Current directory	Directory in which the shell/process is working.	pwd shows it
Parent directory	Directory one level above current directory.	..

8. Exam-Oriented Comparison Tables

Topic	Key differences
Sequential vs Direct	Sequential processes data in order; direct access jumps to a required position.
Contiguous vs Linked	Contiguous uses consecutive blocks; linked uses a chain of blocks anywhere.
Linked vs Indexed	Linked stores next-block pointers; indexed stores block addresses in an index block.
Single vs Two Level	Single has one directory; two-level separates user directories.
Two Level vs Tree	Two-level gives limited user-wise separation; tree provides multi-level hierarchy.

9. Question-Paper Mapping: Unit V

Paper	Mapped Unit V questions
Summer 2026	Q1(g) direct access; Q2(d) tree-structured directory; Q3(d) file operations; Q4(e) disadvantages of contiguous allocation.
Winter 2025	Q1(f) file operations; Q2(d) two-level directory; Q3(b) file types/extensions; Q3(c) linked vs indexed allocation.

Past-paper observation Across the two supplied papers, Unit V produced 8 sub-question opportunities. This is an observation of the supplied papers, not a prediction of future questions.

10. Exam Checklist

- Define a file and explain file attributes.
- Explain common file operations.
- List file types with examples/extensions and explain the purpose of an extension.
- Explain file-system structure with a simple diagram.
- Explain sequential and direct access methods.
- Explain contiguous, linked and indexed allocation.
- Compare linked and indexed allocation using at least four points.
- Explain single-level, two-level and tree-structured directories.
- Draw and explain a tree directory.
- Explain basic file/directory permissions and commands such as mkdir and chmod.

11. Quick Revision Sheet

Topic	One-line memory aid
File	Named collection of related information.
Attribute	Metadata describing the file.
Create/Open/Read/Write/Close	Core file operations.
Sequential access	Read/process in order.
Direct access	Jump to required position.
Contiguous	Consecutive blocks.
Linked	Each block points to next.
Indexed	Index block stores addresses.
Single-level	One directory.
Two-level	Separate user directories.
Tree	Hierarchical directories.
r/w/x	4/2/1 permissions.
754	Owner rwx, group r-x, others r--.

12. References and Research Basis

- MSBTE, Operating System, Course Code 315319, Semester V, K Scheme — Unit V File Management and TLO 5.1–5.3.
- Dhananjay M. Dhamdhere, Operating System: A Concept-Based Approach, 3rd ed., McGraw Hill Education.
- William Stallings, Operating Systems: Internals and Design Principles, 9th ed., Pearson.
- Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, 10th ed., Wiley — File-System Interface and File-System Implementation chapters.