

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

4/17/2025







9

9

10

- UNIT I OVERVIEW AND PROCESS MANAGEMENT
- 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
- 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
- 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
- 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
- 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

9

4/17/2025



FILE SYSTEMS



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



File Types – Name, Extension



- Different file types are supported by operating systems.
- Each file type can have different extensions.
- Figure shows examples of different file types and different extensions for each file type.
- For example a source code file can have the extensions .c or .cc or .java and so on.



File Types – Name, Extension



| file type | usual extension | function |
|----------------|-----------------------------------|------------------------------------------------------------------------------------------------|
| executable | exe, com, bin or none | read 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, rrf, doc | various word-processor formats |
| library | lib, a, so, dll, mpeg, mov, rm | libraries of routines for programmers |
| print or view | arc, zip, tar | ASCII or binary file in a format for printing or viewing |
| archive | arc, zip, tar | related files grouped into one file, sometimes com- pressed, for archiving or storage |
| multimedia | mpeg. mov. rm | binary file containing audio or A/V information |



File Access Methods



- Information stored in files must be accessed.
- There are different methods to access files.
- Some systems support only one method.
- Some systems support multiple methods and the right one is chosen based on the application.
- The different methods of file access are sequential access, direct access and indexed sequential access.
- We learn each of these methods in this section one after the other.



Sequential Access



- The sequential access is the simplest and the most common file access method. In this method, the contents of a file are accessed one after the other. That is, information is processed one record after another. The possible operations can be
- read next read the next record
- write next write the next record
- reset reset the file pointer to the initial record





Sequential Access



- Figure shows the possible operations using sequential access method.
- The current file position is shown.
- From the current file position, it is possible to read the next record or write the next record or rewind back to the beginning of the file.
- Using the sequential access method, it is not possible to access any disk block in random.





- The direct access method is based on the disk model of the file.
- File contents are stored in disk blocks in the disk.
- Disks allow access to any random block.
- Direct access also allows arbitrary blocks to be read and written.





- With direct access, the following operations are possible:
- read n read the nth block
- write n write to the nth block
- position to n move the pointer to the nth block
- read next read the next block after moving to the nth block
- write next write to the next block after moving to the nth block
- rewrite n rewrite the nth block

4/17/2025





- where n is the block number relative to beginning of file
- it is possible to implement sequential access using the operations used for direct access.
- Figure shows the sequential access operations and the corresponding operations in direct access method.
- Thus, it is possible to simulate sequential access on a direct file.
- But, direct access cannot be simulated on a sequential access file.





| sequential access | implementation for direct access |
|-------------------|----------------------------------|
| reset | cp = 0; |
| read next | read cp ; cp = cp+1; |
| write next | write cp; cp = cp+1; |



Other Access Methods



- In another access method, an index is built which contains pointers to various blocks of the file.
- To find a record, the index is searched, and the pointer is used to access the file and the desired record.
- Searching through an index is faster.
- When the file size becomes large, the index file also becomes large.
- Therefore, to make the access faster, another secondary index can be maintained for the primary index file.
- The primary index file will have pointers to secondary index files, and the secondary index files point to blocks of the file.
- Figure shows an example of how an index file is used.

4/17/2025

