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

Syllabus

- UNIT I OVERVIEW AND PROCESS MANAGEMENT 9
- 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 10
- 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 9
- 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 8
- 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 9
- 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

FILE SYSTEMS

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

File-System Mounting

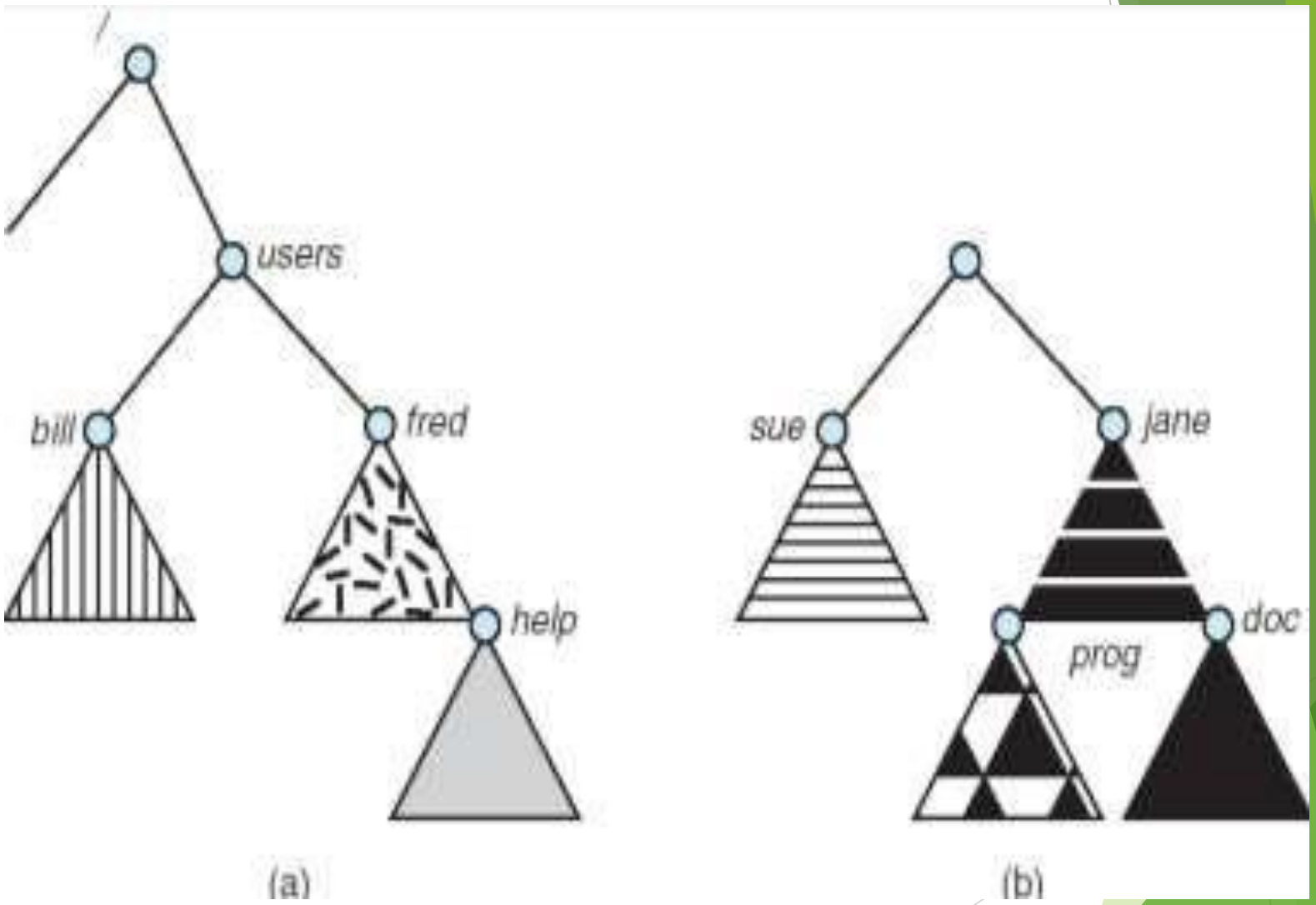
- The file system is the most visible part of an operating system.
- File systems are kept in secondary storage devices.
- File systems can be kept in different partitions, different disks, pen drives and so on.
- In this module we will learn how a file system can be mounted on another file system, how sharing of files is supported in file systems and how files can be protected.

File-System Mounting

- Similar to how a file must be opened before use, a file system must be mounted before it can be accessed.
- A single file system can be built out of multiple partitions or there can be a separate file-system in each partition.
- To logically attach different file systems together and to view the files of different file systems, mounting is done.

- The mount procedure is as follows:
- There is a *mount* command/system call which is used for mounting.
- The system is provided with the name of device containing the file system to be mounted and the mount point as arguments to the mount call.
- The mount point is the directory at which the mounted file system will be attached.

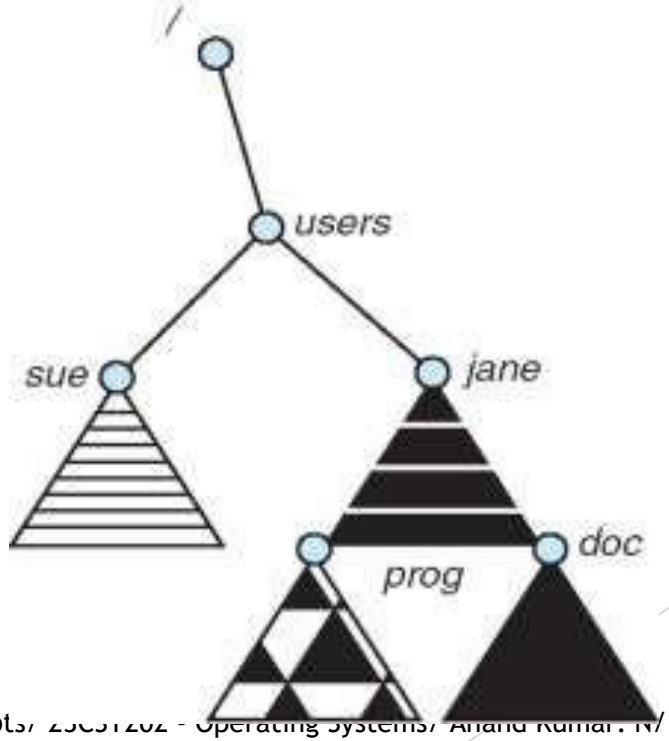
File-System Mounting



(a) Existing file system. (b) Unmounted file system

File Sharing

- Sharing of files is desirable when users want to collaborate and want to achieve a computing goal.
- Therefore, operating systems must provide support to share files, in spite of the difficulties.
- In this section, we will discuss different issues that may arise when files are shared.



File Sharing – Multiple Users

- In a single user system, the need for the sharing of files may not arise.
- But, in a multiuser system, more protection and access control are needed for file sharing.
- Therefore, the system must maintain more file and directory attributes than are needed in a single-user operating system.
- Most systems use the concepts of owner (user) and group.
- The owner is responsible for changing attributes and granting access and has the most control over the file.
- A group defines the subset of users who can share access to the file.

File Sharing – Multiple Users

- For example, in UNIX, the owner of a file can issue all operations on a file like reading, writing and executing.
- A group can have permissions to issue a subset of operations and all others (other users) can have different access permissions.
- The owner and group are assigned IDs and the owner and group IDs are stored along with the other attributes of the file.
- Even with multiple local file systems, ID checking and permission matching are straightforward, once the file systems are mounted.
- But, what happens when the file systems are not local, but placed in different locations connected through a network?

File Sharing – Remote File Systems

- With the advent of computer networks, communication among remote computers became possible.
- Networking allows the sharing of resources spread across the world.
- Data in the form of files is one such resource.

- Files can be shared using the following methods:
- In the first method, the files are transferred manually via programs like FTP.
- In the second method, a distributed file system is used, in which remote directories are visible from a local machine In the third method, the World Wide Web is used.
- A browser is used to gain access to the remote files.
- The World Wide Web uses anonymous file exchange.

File Sharing – Remote File Systems

- The client-server model allows clients to mount remote file systems from servers.
- A server can serve multiple clients and a client can use multiple servers.
- The NFS is a standard UNIX client-server file sharing protocol.
- The Common Internet File System (CIFS) is standard Windows protocol.

File Protection

- It is necessary to keep files safe from physical damage (reliability) and from improper access (protection).
- For keeping the files reliable, it is necessary to have duplicate copies of files.
- We can also periodically copy disk files to tape at regular intervals.

- We now see how files can be protected from improper access.
- The owner/creator of the file should be able to control what can be done on a file and by whom.
- The types of access that can be controlled are read, write, execute, append, delete, list, renaming, copying etc.

- One way in which access can be controlled is to have access control lists and groups.

Access Control Lists and Groups

- With each file and directory an access-control list (ACL) is attached.
- The ACL has the names of users and the types of access allowed for each user.
- When a user requests access to a particular file, the access list is checked.
- If the user is listed for that particular access, access is allowed. Else, user is denied access.

Access Control Lists and Groups

- With each file and directory an access-control list (ACL) is attached.
- The ACL has the names of users and the types of access allowed for each user.
- When a user requests access to a particular file, the access list is checked.
- If the user is listed for that particular access, access is allowed. Else, user is denied access.