



# OPERATING SYSTEMS

## Storage Management - 11

**Nitin V Pujari**  
**Faculty, Computer Science**  
**Dean - IQAC, PES University**



**Nitin V Pujari**  
**Faculty, Computer Science**  
**Dean - IQAC, PES University**

# OPERATING SYSTEMS

**Storage Management - 11:**  
**Case Study: Network File System - NFS**

---

# OPERATING SYSTEMS

## Course Syllabus - Unit 4

---



### Unit 4: Storage Management

Mass-Storage Structure - Mass-Storage overview, Disk Scheduling, Swap-Space Management, RAID structure. File System Interface - file organization/structure and access methods, directories, sharing File System Implementation/Internals: File control Block (inode), partitions & mounting, Allocation methods.

Case Study: Linux/Windows File Systems

# OPERATING SYSTEMS

## Course Outline



37	Mass-Storage Structure: Mass-Storage overview	12.1	82.1
38	Disk Scheduling – FCFS, SSTF, SCAN, C-SCAN, LOOK	12.4	
39	Swap-Space Management, RAID Structure	12.6,12.7	
40	File Concept, File Structure, Access Methods	10.1-10.2	
41	Directory and Disk Structure	10.3	
42	File-System Mounting, File Sharing, Protecting	10.4-10.6	
43	Implementing File-Systems: File control Block (inode), partitions & mounting	11.1,11.2	
44	Disk Space Allocation methods: Contiguous, Linked, Indexed	11.4	
45	Case Study: Unix/Linux File systems	16.7	
46	NFS	11.8	

- **Network File System -  
NFS**

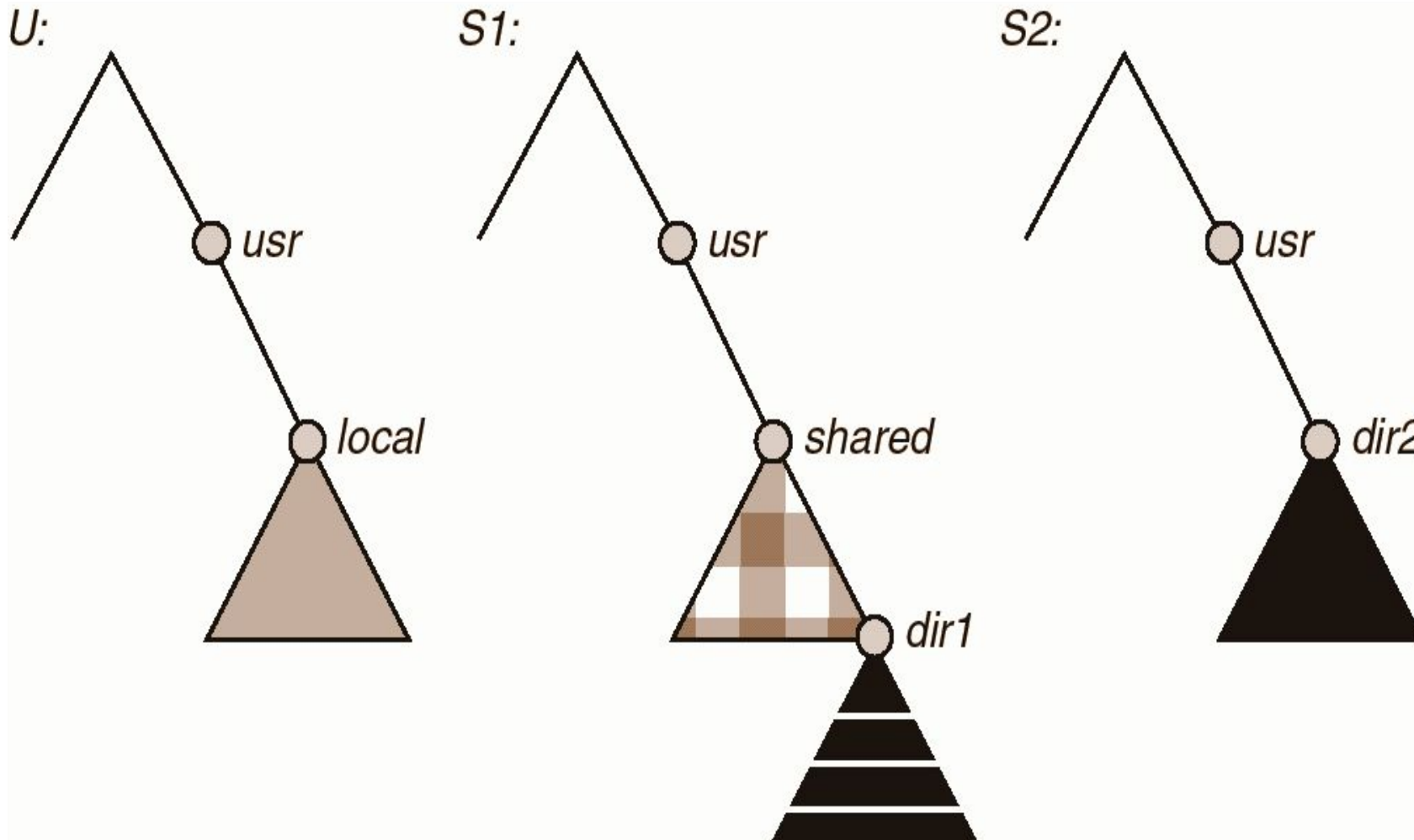
- An implementation and a specification of a software system for accessing remote files across LANs (or WANs)
- The implementation is part of the Solaris and Sun operating systems running on Sun workstations using an Unreliable / User Datagram Protocol and Ethernet Internetwork Protocol (**UDP/IP** protocol)

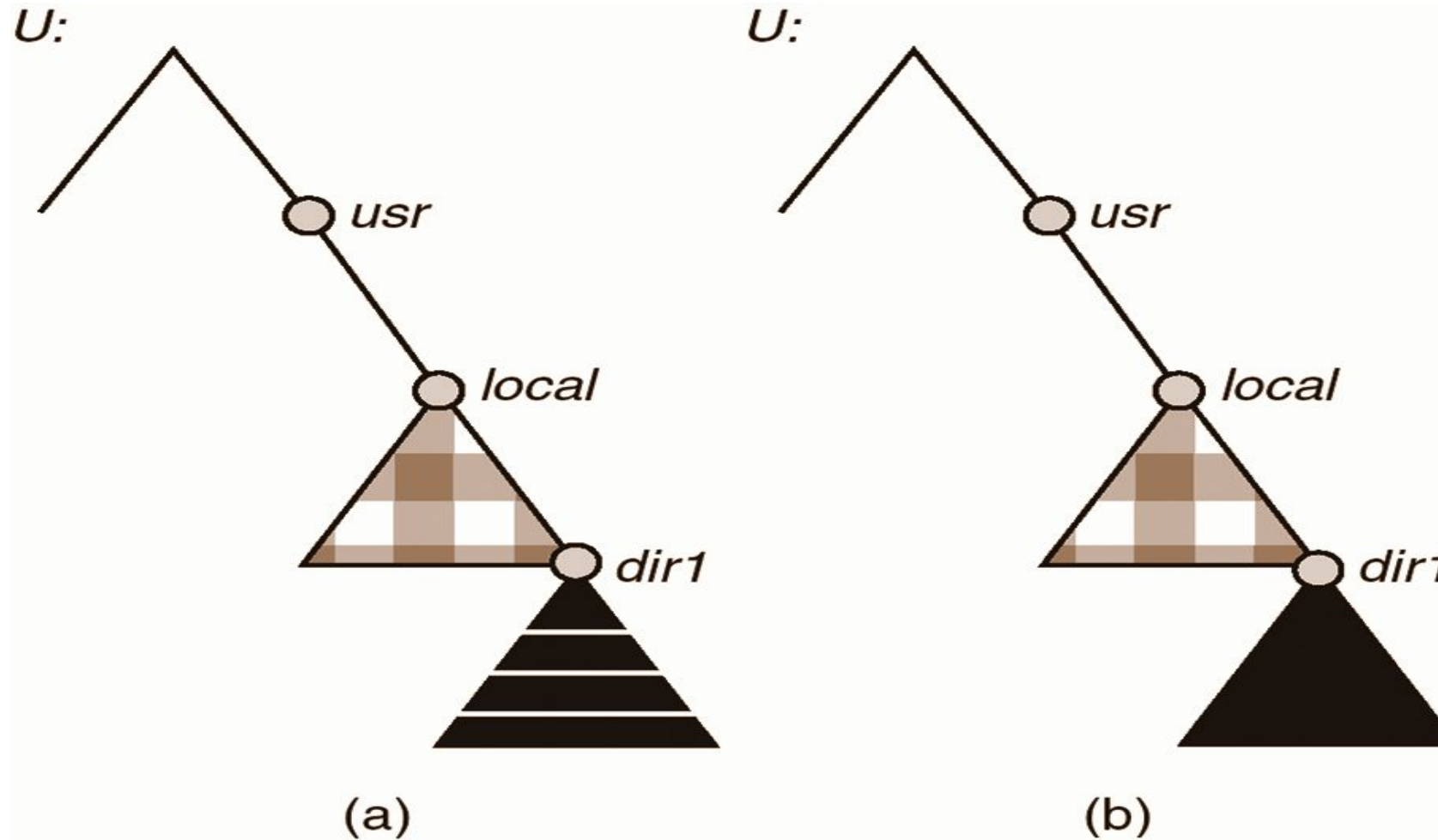
- Interconnected workstations viewed as a set of independent machines with independent file systems, which allows sharing among these file systems in a transparent manner
  - A remote directory is mounted over a local file system directory
    - The mounted directory looks like an integral subtree of the local file system, replacing the subtree descending from the local directory
  - Specification of the remote directory for the mount operation is non-transparent; the host name of the remote directory has to be provided
    - Files in the remote directory can then be accessed in a transparent manner
- Subject to access-rights accreditation, potentially any file system (or directory within a file system), can be mounted remotely on top of any local directory

- NFS is designed to operate in a heterogeneous environment of different machines, operating systems, and network architectures; the NFS specifications independent of these media
- This independence is achieved through the use of RPC primitives built on top of an External Data Representation (XDR) protocol used between two implementation independent interfaces
- The NFS specification distinguishes between the services provided by a mount mechanism and the actual remote-file-access services



## Three Independent File Systems





- Establishes initial logical connection between server and client
- Mount operation includes name of remote directory to be mounted and name of server machine storing it
  - Mount request is mapped to corresponding RPC and forwarded to mount server running on server machine
  - Export list – specifies local file systems that server exports for mounting, along with names of machines that are permitted to mount them

- Following a mount request that conforms to its export list, the server returns a file handle—a key for further accesses
- File handle – a file-system identifier, and an inode number to identify the mounted directory within the exported file system
- The mount operation changes only the user's view and does not affect the server side

## Network File System protocol

---

- Provides a set of remote procedure calls for remote file operations. The procedures support the following operations:
  - searching for a file within a directory
  - reading a set of directory entries
  - manipulating links and directories
  - accessing file attributes
  - reading and writing files

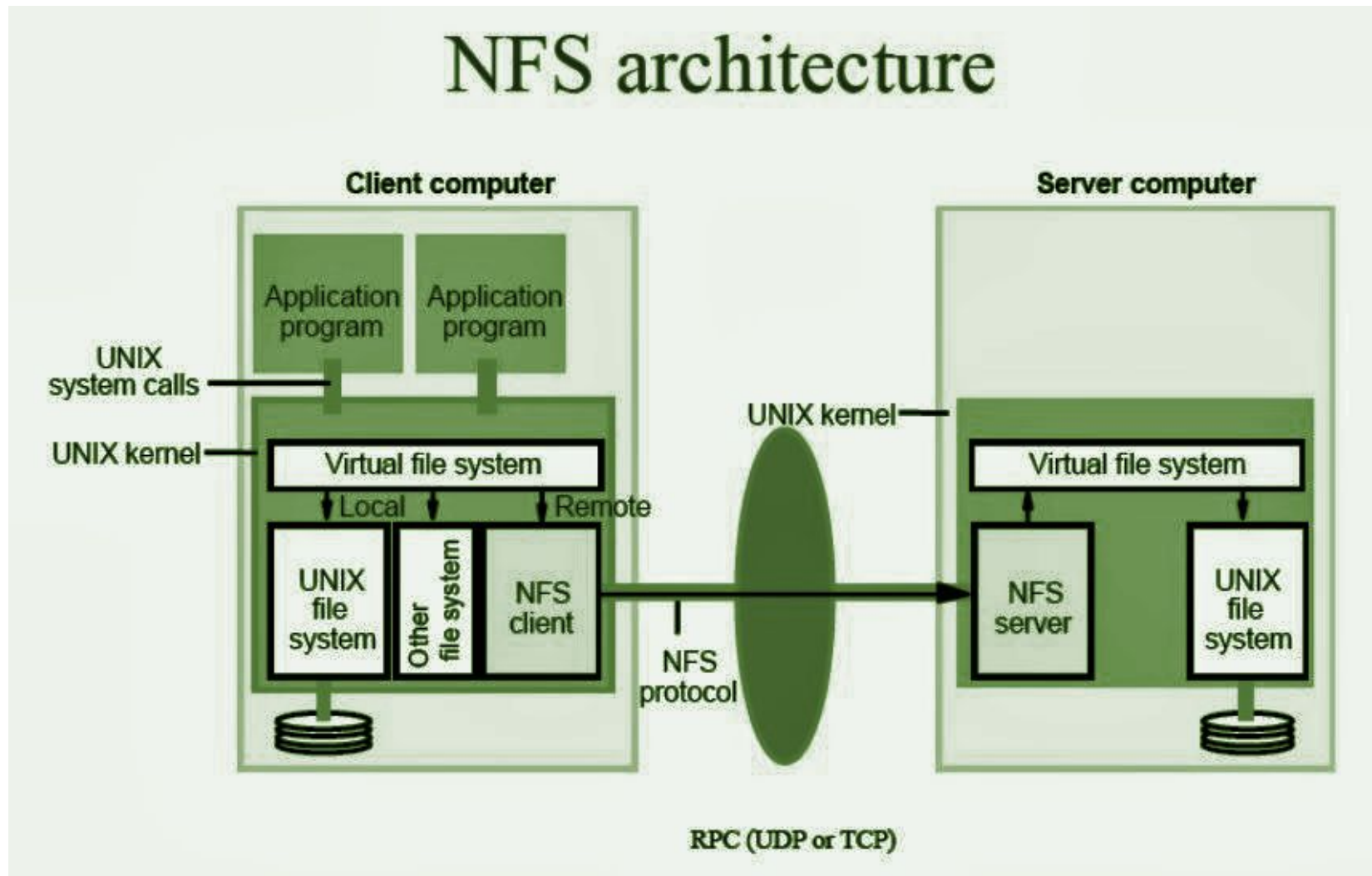
- NFS servers are **stateless**; each request has to provide a full set of arguments (NFS V4 is just coming available – very different, stateful)
- Modified data must be committed to the server's disk before results are returned to the client (lose advantages of caching)
- The NFS protocol does not provide concurrency-control mechanisms

## Three Major Layers of Network File System Architecture

---

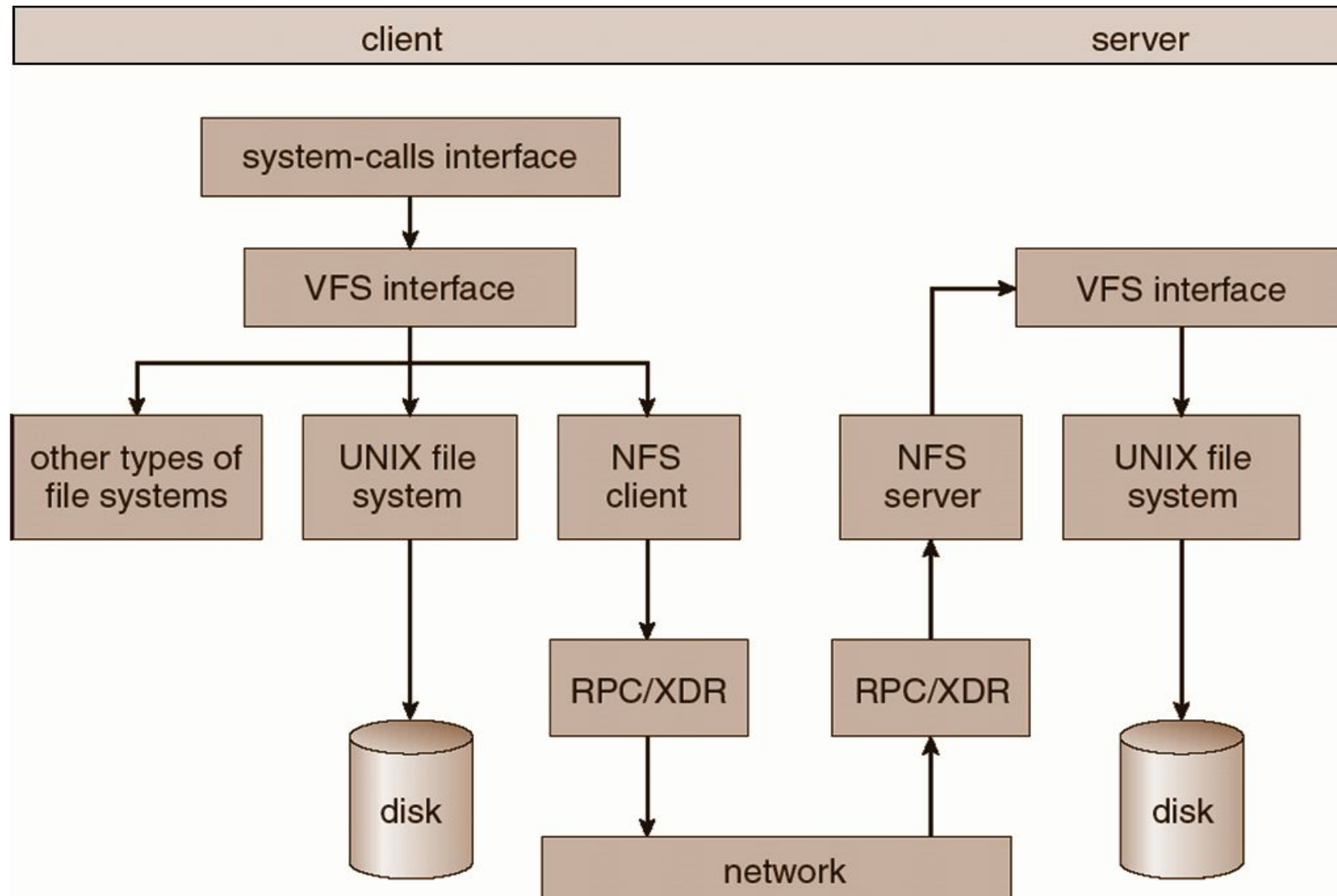


- UNIX file-system interface (based on the **open**, **read**, **write**, and **close** calls, and **file descriptors**)
- Virtual File System (VFS) layer – distinguishes local files from remote ones, and local files are further distinguished according to their file-system types
  - The VFS activates file-system-specific operations to handle local requests according to their file-system types
  - Calls the NFS protocol procedures for remote requests
- NFS service layer – bottom layer of the architecture
  - Implements the NFS protocol





## Schematic View of Network File System Architecture



# OPERATING SYSTEMS

## Typical Network File System Dashboard



Server Manager

← → ▾ Server Manager ▸ Dashboard

Manage Tools View Help

Dashboard

Local Server

All Servers

File and Storage Services ▸

WELCOME TO SERVER MANAGER

QUICK START

WHAT'S NEW

LEARN MORE

1 Configure this local server

2 Add roles and features

3 Add other servers to manage

4 Create a server group

Hide

ROLES AND SERVER GROUPS

Roles: 1 | Server groups: 1 | Servers total: 1

File and Storage Services 1

Manageability

Events

Services

Performance

BPA results

Local Server 1

Manageability

Events

Services

Performance

BPA results

Add Roles and Features

Remove Roles and Features

Add Servers

Create Server Group

Server Manager Properties

- Performed by breaking the path into component names and performing a separate NFS lookup call for every pair of component name and directory vnode
- To make lookup faster, a directory name lookup cache on the client's side holds the vnodes for remote directory names

# Network File System Remote

---

## Operations

- Nearly one-to-one correspondence between regular UNIX system calls and the NFS protocol RPCs (except opening and closing files)
- NFS adheres to the remote-service paradigm, but employs buffering and caching techniques for the sake of performance

# Network File System Remote

---

## Operations

- File-blocks cache – when a file is opened, the kernel checks with the remote server whether to fetch or revalidate the cached attributes
  - Cached file blocks are used only if the corresponding cached attributes are up to date
- File-attribute cache – the attribute cache is updated whenever new attributes arrive from the server
- Clients do not free delayed-write blocks until the server confirms that the data have been written to disk

- **Network File System -  
NFS**



## **THANK YOU**

**Nitin V Pujari**  
**Faculty, Computer Science**  
**Dean - IQAC, PES University**

**nitin.pujari@pes.edu**

**For Course Deliverables by the Anchor Faculty click on [www.pesuacademy.com](http://www.pesuacademy.com)**