

# Distributed Systems

---

COMP90015 2023 Semester 1  
Tutorial 06

# Today's Agenda

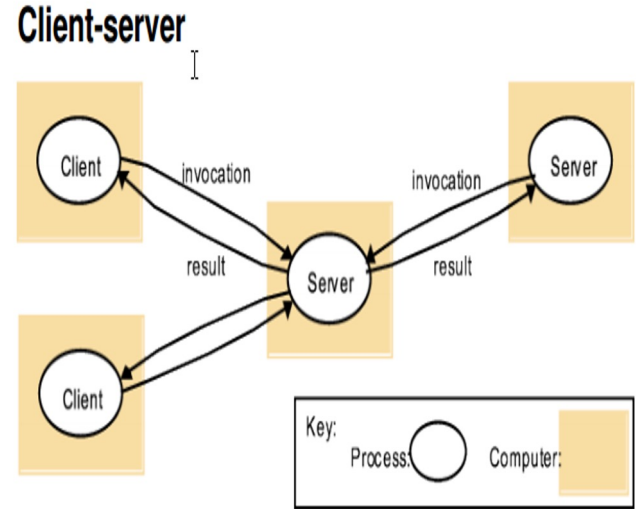
Discussion on open questions on Distributed Systems models

Operating Systems Support Questions

Q1. Briefly explain the difference between a client-server architecture and a peer-to-peer architecture.

# Client- Server

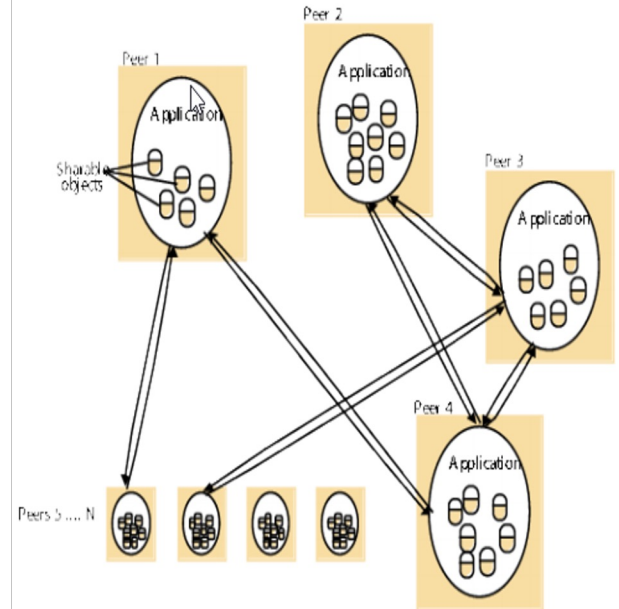
- A client requests some processing or information from a server that it needs.
- It waits in a blocking fashion for the reply containing the result,
- It then can proceed with it's execution
- There can be many variants of client server model



# Peer-to-Peer

- Peer model suits ad-hoc groupings of participants
- No **central point of failure** (reliable)
- No **central point of control** (difficult to deny service for adversaries)
- Some peers will typically **contribute more than others** (i.e. seed or super-peer)
- Examples- Napster, BitTorrent

## Peer-to-Peer



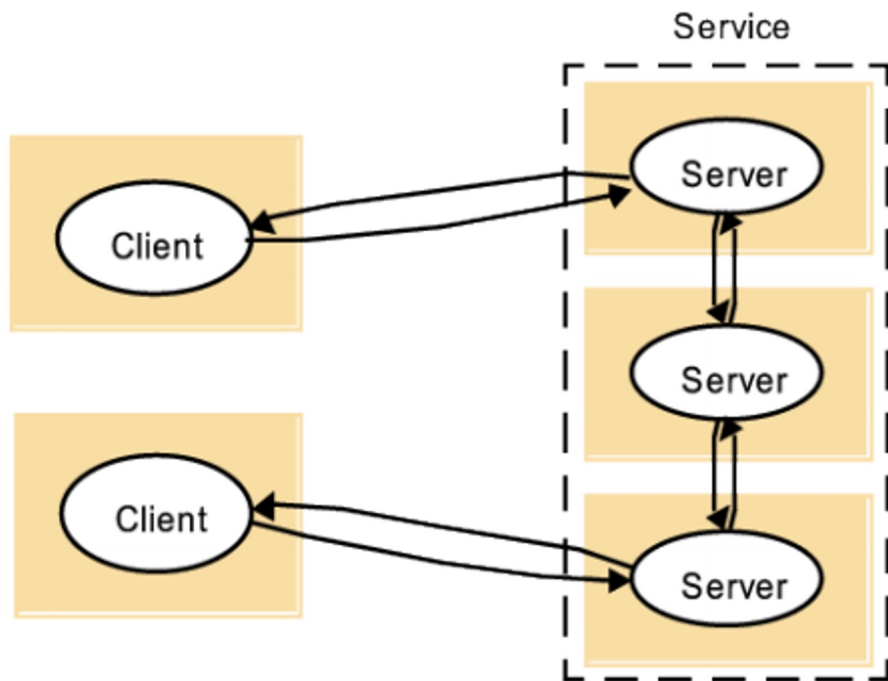
Q2. Briefly explain each of the following distributed system architecture variations, giving also a reason or a benefit of its use:

- Services provided by multiple servers
- Proxy servers and caches
- Mobile code and Mobile Agents
- Network computers
- Thin clients
- Tiered Architecture

- Services provided by multiple servers
- Proxy servers and caches
- Mobile code and Mobile Agents
- Network computers
- Thin clients
- Tiered Architecture

## A service provided by multiple servers

Service is provided by several server processes interacting with each other. Objects may be partitioned (e.g. web servers) or replicated across servers (e.g. Sun Network Information Service (NIS)).

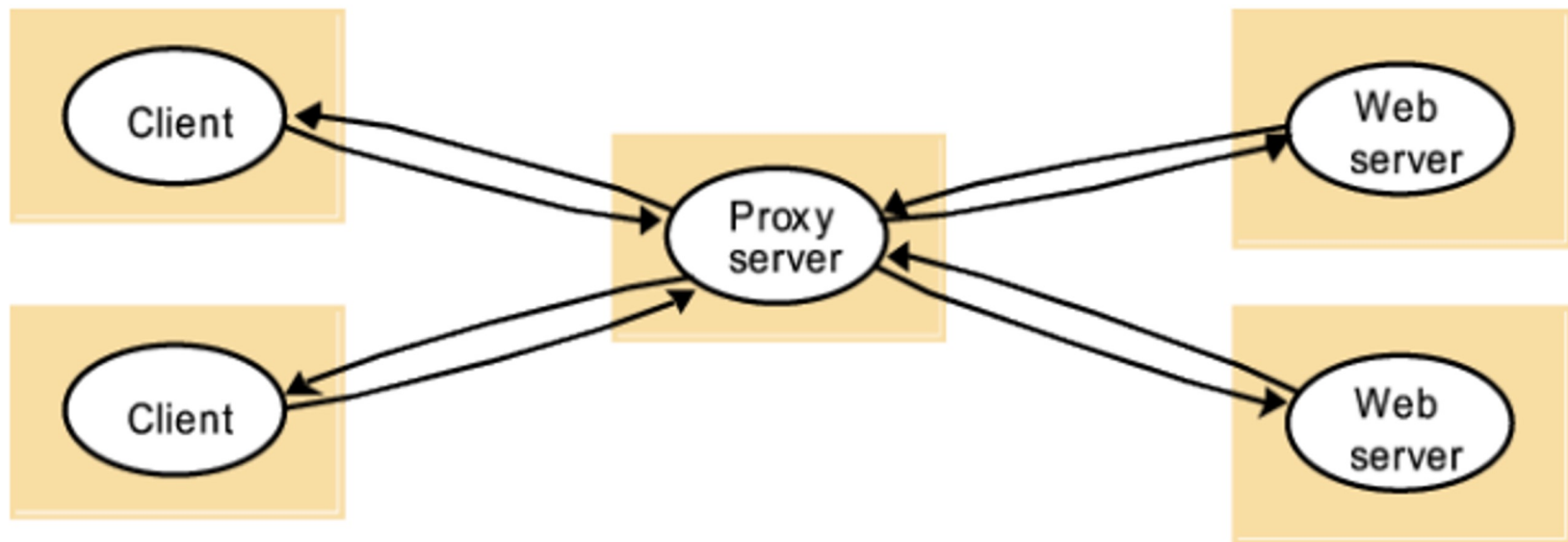




- Services provided by multiple servers
- **Proxy servers and caches**
- Mobile code and Mobile Agents
- Network computers
- Thin clients
- Tiered Architecture

# Proxy servers and caches

- Cache is a store of recently used objects that is closer to client
- New objects are added to the cache replacing existing objects
- When an object is requested, the caching service is checked to see if an up-to-date copy is available (fetched if not available)



- Services provided by multiple servers
- Proxy servers and caches
- **Mobile code and Mobile Agents**
- Network computers
- Thin clients
- Tiered Architecture

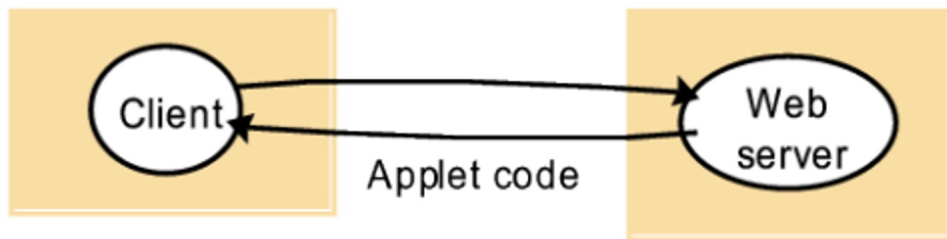
# Mobile Code and Agents

*Mobile Code* is down loaded to the client and is executed on the client (e.g. applet).

*Mobile agents* are running programs that includes both code and data that travels from one computer to another.

E.g. Web Applets:

a) client request results in the downloading of applet code



b) client interacts with the applet

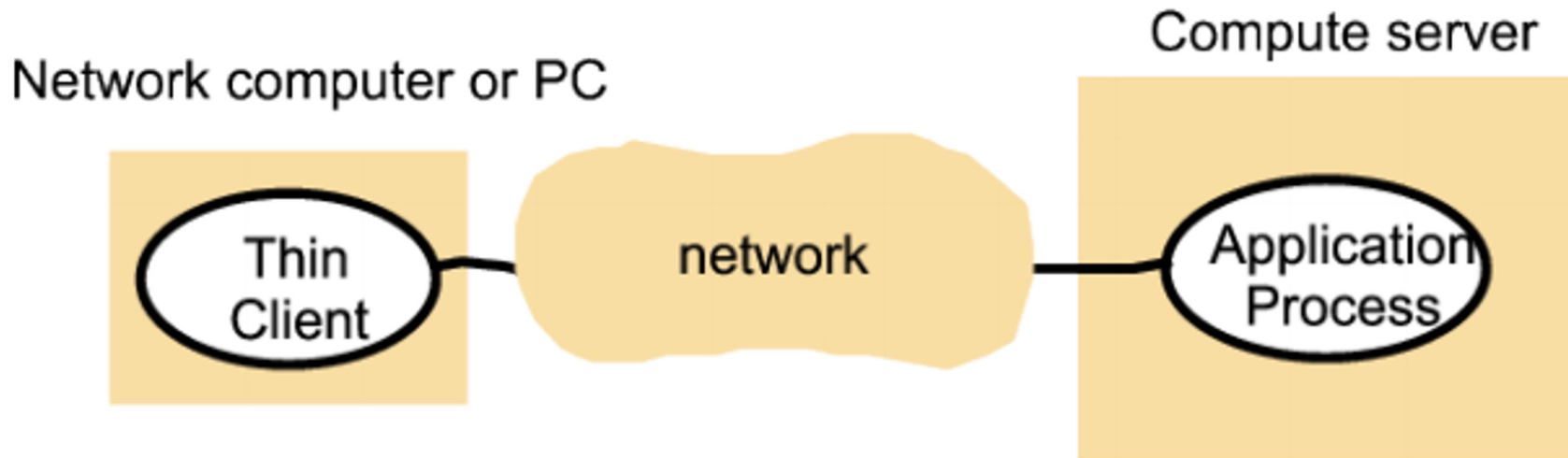


- Services provided by multiple servers
- Proxy servers and caches
- Mobile code and Mobile Agents
- Network computers
- Thin clients
- Tiered Architecture

# Network Computers and Thin clients

- **Network Computers:** download their operating system and application software from a remote file system. Applications are run locally.
- **Thin Clients:** application software is not downloaded but runs on the computer server - e.g. UNIX.

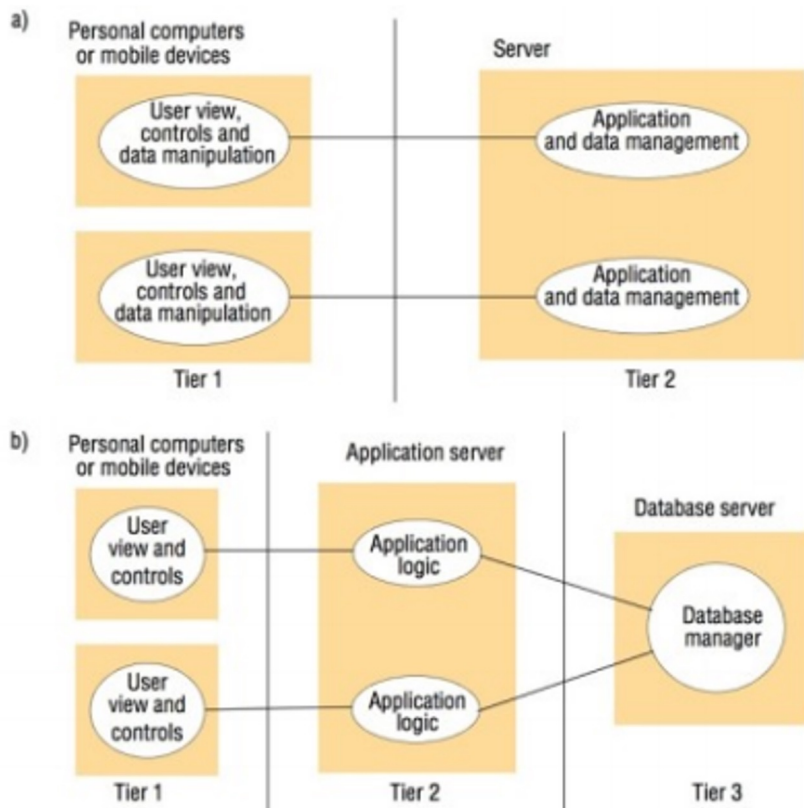
This paradigm is usually not suitable for highly interactive graphical activities.



- Services provided by multiple servers
- Proxy servers and caches
- Mobile code and Mobile Agents
- Network computers
- Thin clients
- Tiered Architecture

# Tiered architecture

Tiered architectures are complementary to layering. Layering deals with vertical organization of services.





# Operating Systems Support Questions

1. Discuss the difference between a network operating system and a distributed operating system.

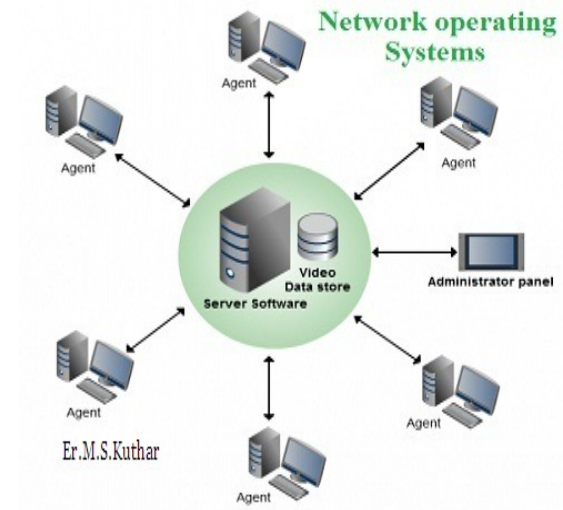
1. Discuss the difference between a network operating system and a distributed operating system.

- A *networked operating system* provides **support for networking operations**. The users are generally expected to make intelligent use of the network commands and operations that are provided. Each host **remains autonomous** in the sense that it can continue to operate when disconnected from the networking environment.
- A *distributed operating system* tries to **abstract the network from the user** and thereby remove the need for the user to specify how the networking commands and operations should be undertaken. This is sometimes referred to as providing a **single system image**. Each host may not have everything that would be required to operate on its own, when disconnected from the network.

1. Discuss the difference between a network operating system and a distributed operating system.

## Network operating system

- Users retain **autonomy** in managing their own processing resources
- It does **not manage processes across** the nodes
- Provides **support** for networking operations



1. Discuss the difference between a network operation system and a distributed operating system.

### **Distributed operating system**

- Users are never concerned with where their programs run, or the location of any resources
- Has control over all the nodes in the system, and it transparently locates new processes at whatever node suits its scheduling policies
- Each host may not have everything that would be required to operate on its own
- Single system image

Helper Link: <https://www.youtube.com/watch?v=NYBKXzl5bWU>

2. What are the core OS components?

## 2. What are the core OS components?

- **Process manager** -- Handles the creation of processes, which is a unit of resource management, encapsulating the basic resources of memory (address space) and processor time (threads).
- **Thread manager** -- Handles the creation, synchronization and scheduling of one or more threads for each process. Threads can be scheduled to receive processor time.
- **Communication manager** -- Handles interprocess communication, i.e. between threads from different processes. In some cases this can be across different hosts

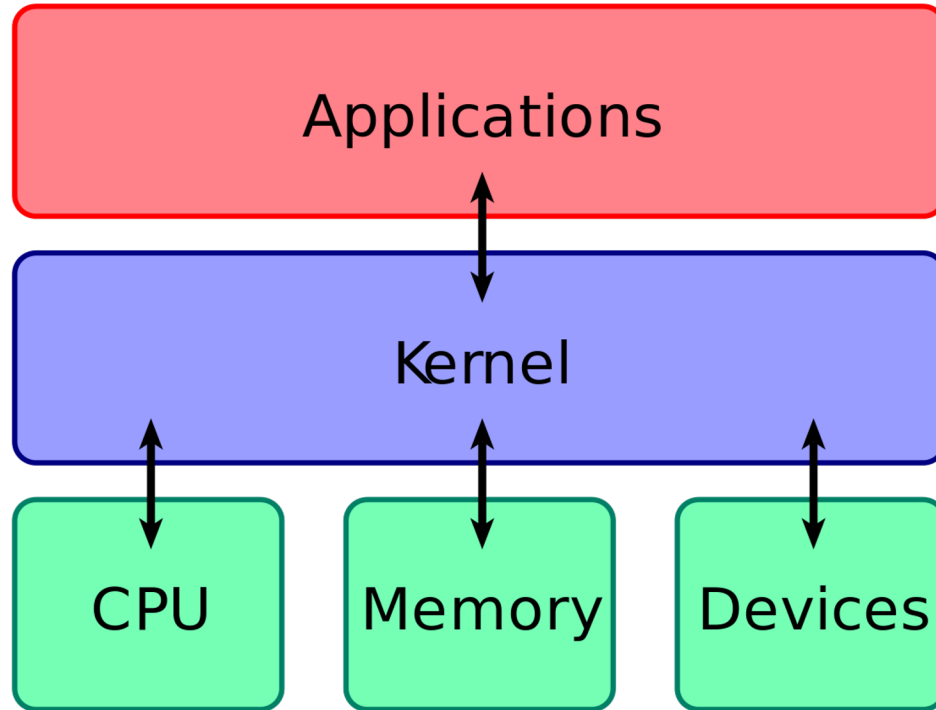
## 2. What are the core OS components?

- **Memory manager** -- Handles the allocation and access to physical and virtual memory. Provides translation from virtual to physical memory and handles paging of memory.
- **Supervisor** -- Handles privileged operations, i.e. those that directly affect shared resources on the host, e.g. to and from an I/O device. The supervisor is responsible for ensuring that host continues to provide proper service to each client.



3. What is the kernel? and also discuss about popular kernel implementation methods

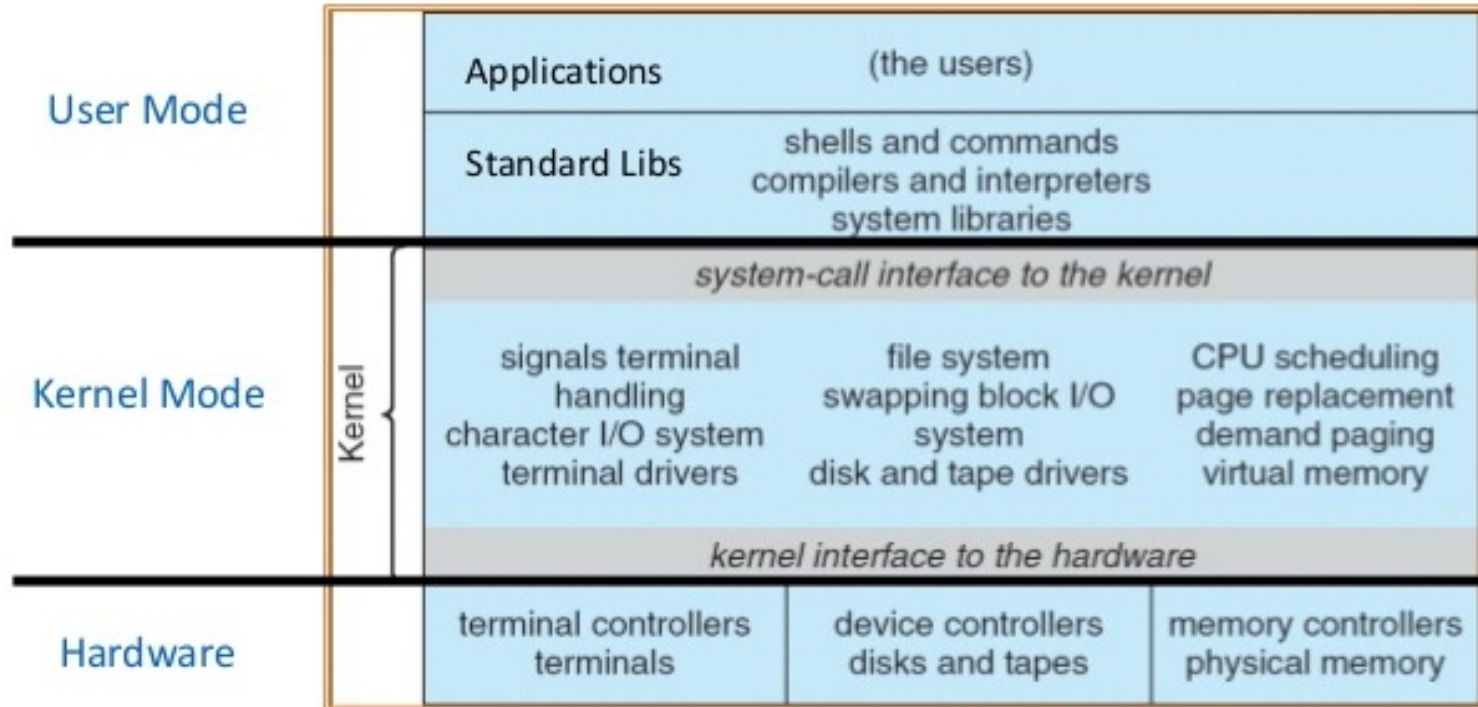
3. What is the kernel? and also discuss about popular kernel implementation methods



### 3. What is the kernel ? and also discuss about popular kernel implementation methods.

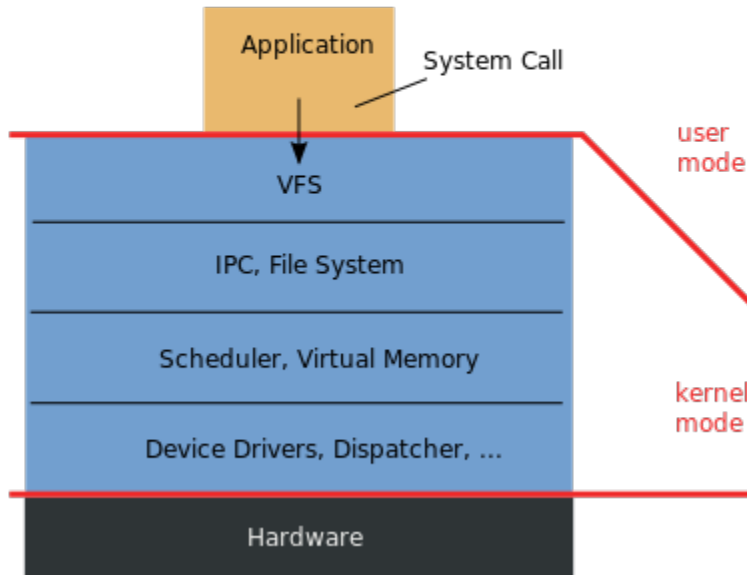
- Part of the Operating System
- Has full access to the host's resources
- Kernel begins execution after the host is powered up and continues to execute while the host is operational
- The kernel has access to all resources and shares access to all other processes that executing on the host

# A Real-World Example - UNIX

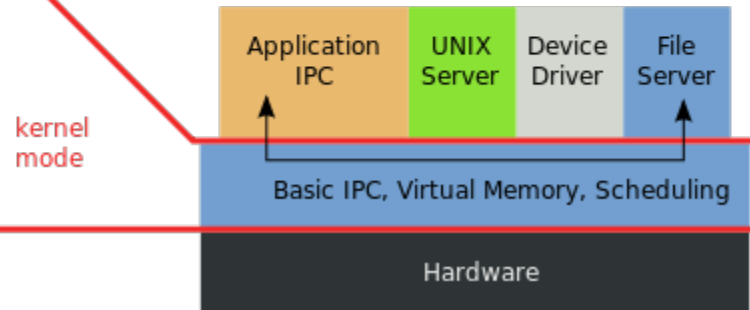


# Comparison of Kernel Implementations

Monolithic Kernel  
based Operating System



Microkernel  
based Operating System



# Disadvantages of Monolithic OS

- It is massive
  - codebase
- It is undifferentiated:
  - non-modular (traditionally), although modern ones are much more layered.
- It is intractable:
  - Altering any individual software component to adapt for new requirements is difficult.

## 4. What is Supervisor Mode vs User Mode?

## 4. What is Supervisor Mode vs User Mode?

- Operating modes supported by the hardware at the machine instruction level.
- **Supervisor / Kernel mode** -- instructions that execute while the processor is in supervisor (or privileged) mode are capable of accessing and controlling every resource on the host,
- **User mode** -- instructions that execute while the processor is in user (or unprivileged) mode are restricted, by the processor, to only those accesses defined or granted by the kernel.