Operating System

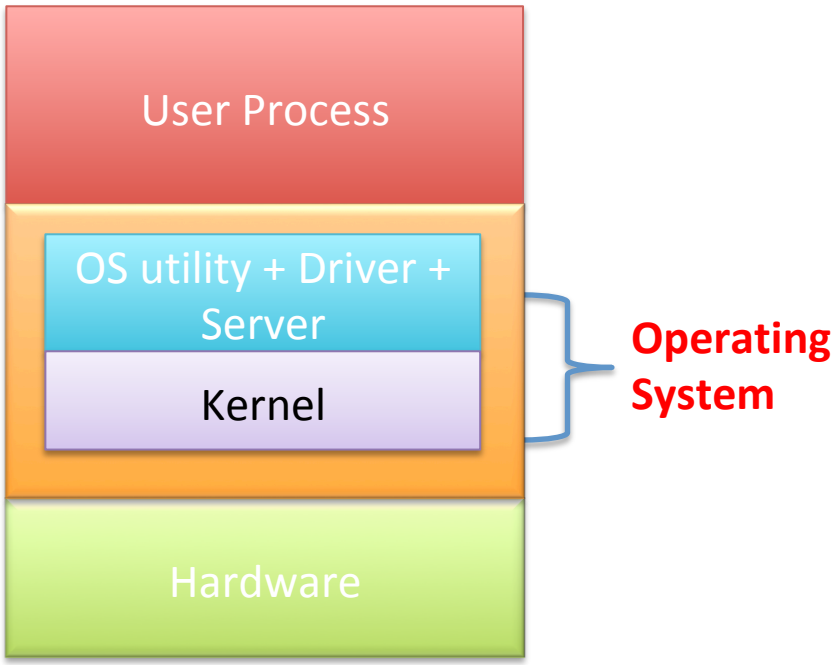
Md. Asifur Rahman
Lecturer
Dept. of CSE, RUET

Operating system

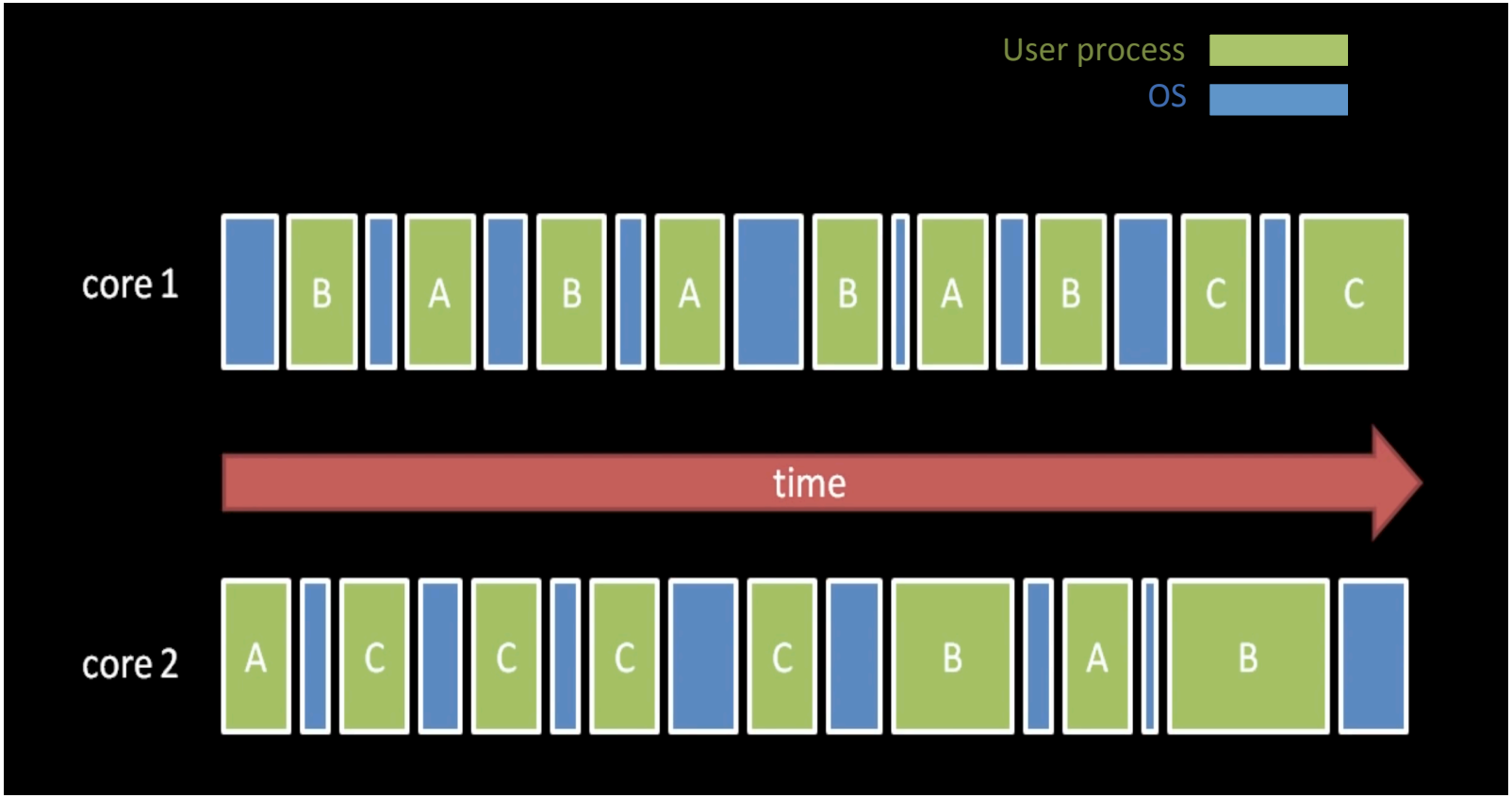
An operating system (OS) is a system software that manages computer hardware and software resources and provide common services for computer programs.

Functions of a general purpose OS :

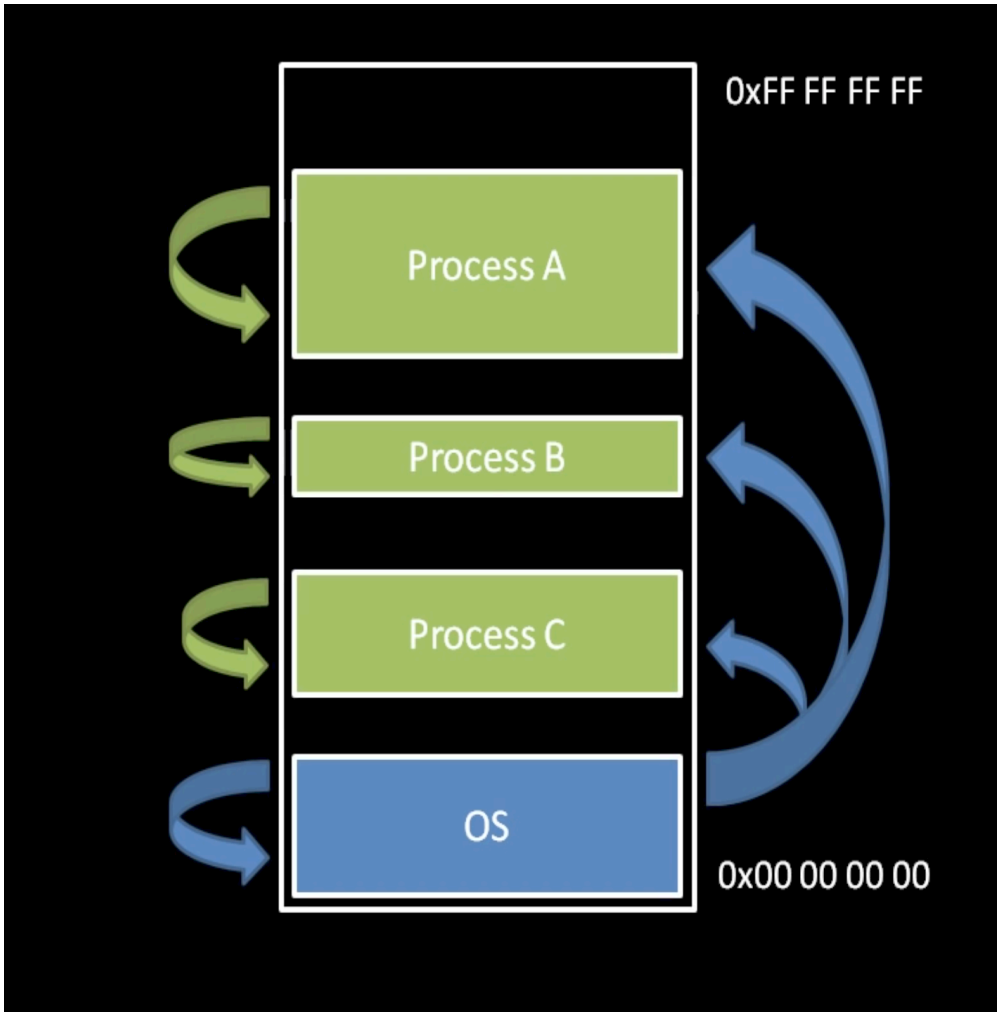
- Coordinate access to physical resources – CPU, memory, i/o devices
- Provide services such as:
  - Protection.
  - Scheduling.
  - Memory Management.
  - File systems.
  - Synchronization.
  - Virtualization.
  - Inter Process Communication
  - User Interface



Operating system



Operating system



OS Need to Protect Usage of

- **CPU :** Use of CPU by any user process. One user process might keep using the CPU forever while depriving other from using it.
- **I/O:** Prevent user process from access illegal I/O or perform illegal I/O command.
- **Memory:**
  - Prevent any user process from accessing another's data
  - Prevent any user process from access and manipulate OS's (Kernel) code and data structure.

Architectural Features to Support OS

- Protected Mode of execution − User mode Vs Kernel Mode.
- Privileged Instructions
- Timer (clock) operation
- Synchronization instruction − test and set
- Memory Protection
- I/O Control Operation
- Interrupts and Exceptions
- Virtualization architecture

Kernel

- Kernel is the central component of an operating system. It contains the privileged instruction as well as other general instructions which allows it to provide core features (through system call) of an OS which are −
  - Process management
  - Device management
  - Memory management
  - Interrupt handling
  - I/O communication

Privileged instruction example

- Change Processor's Mode of Execution – Kernel to User or vice versa

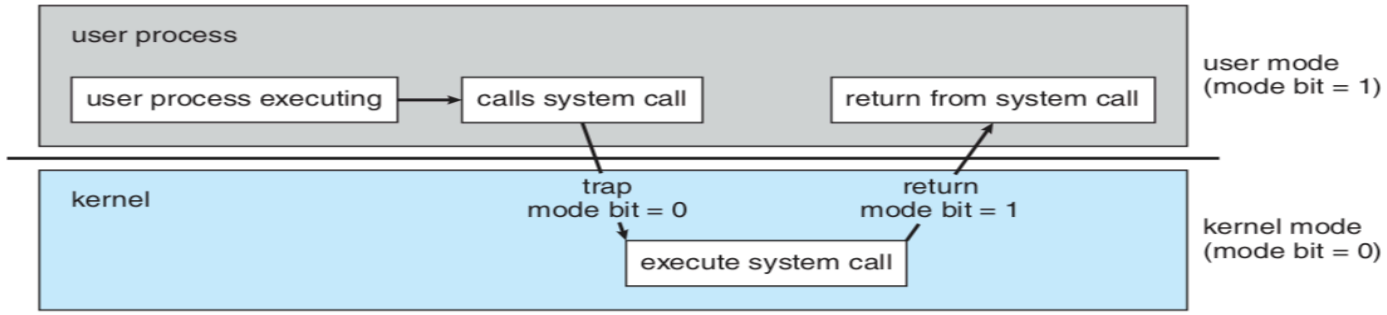
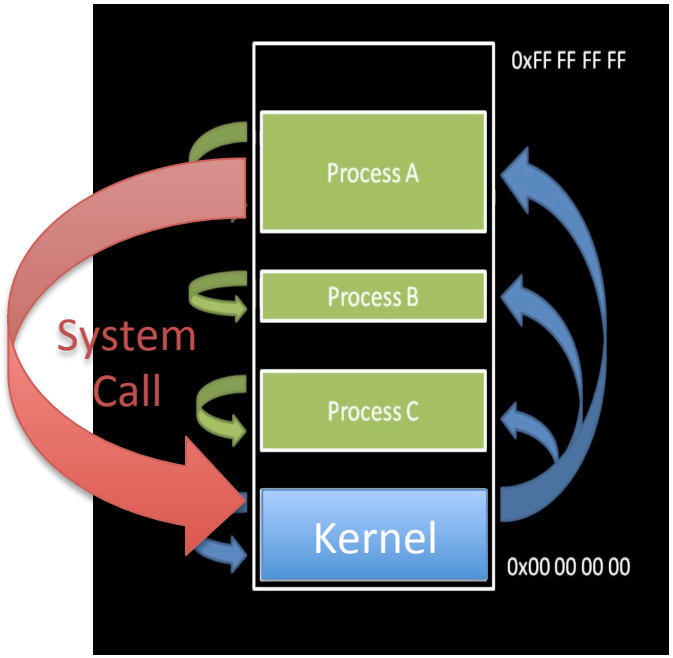
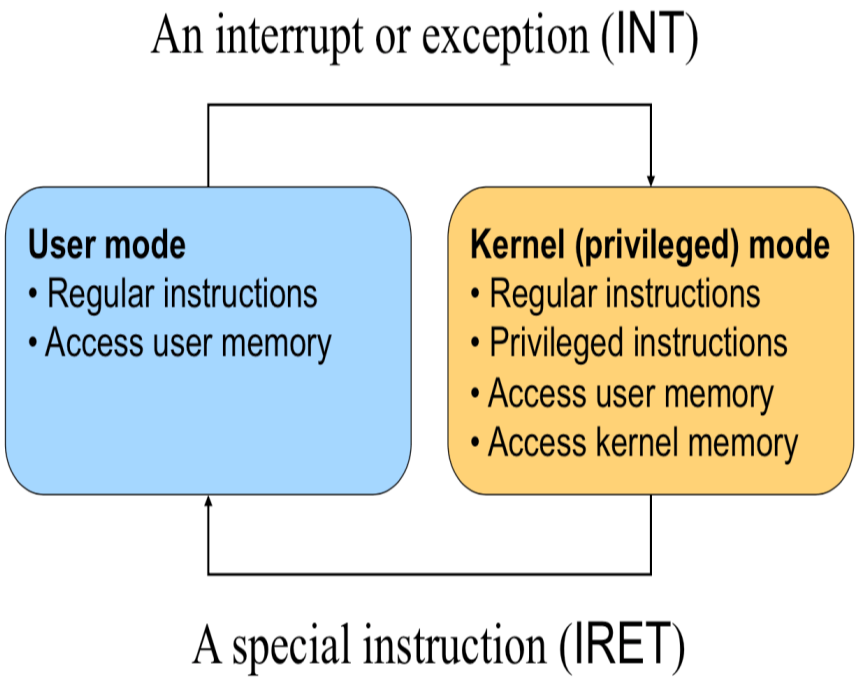


Figure 1.10 Transition from user to kernel mode.



Privileged instruction example contd.

- Load and Read system registers
- Invalidate TLB entries
- Change frequency of clock.
- Reset a processor
- Halt a processor
- Memory address mapping
- Perform I/O operation

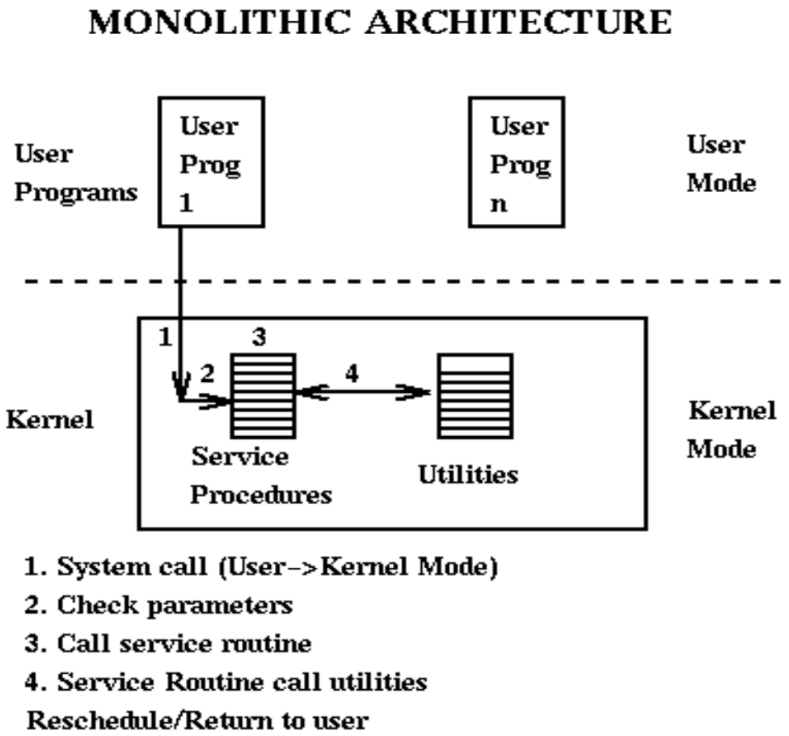


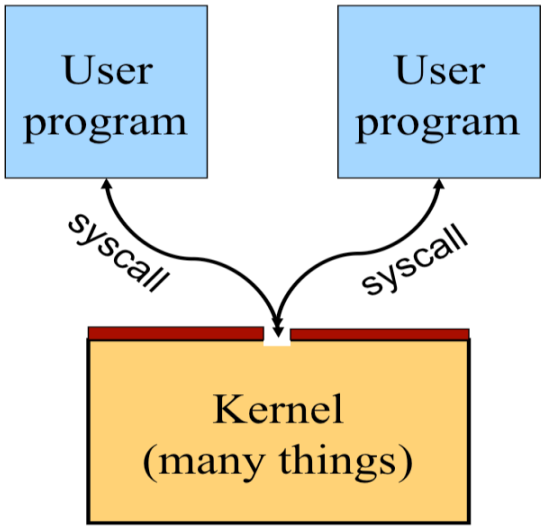
OS Organization

1. Monolithic
2. Layered
3. Microkernel
4. Hybrid
5. Distributed
6. Virtual Machine Based

Monolithic

- The entire operating system runs as a single program in kernel mode. The operating system is written as a collection of procedures, linked together into a single large executable binary program.
- Each procedure in the system is free to call any other one. Hence thousands of procedures can call each other without restriction.





Monolithic

Example : BSD Unix, Windows

Advantage:

- Shared kernel space
- Easy to implement
- Good performance
- Low overheads

Disadvantage:

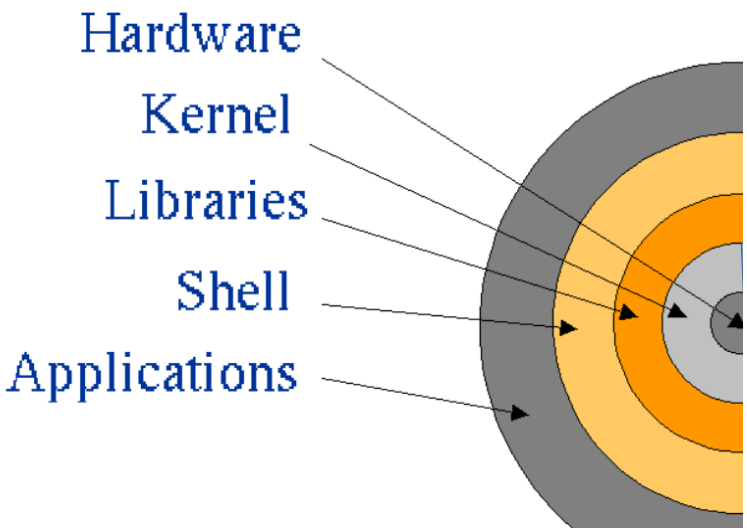
- Instability: crash in any procedure brings system down
- Inflexible / hard to maintain, extend
- One Single executable program.

Layered

- Components are divided into layers
  - Grouping similar components
- Interaction among layer:
  - The outer layer - request service
  - The inner layer – provide service

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Figure 1-25. Structure of the THE operating system.

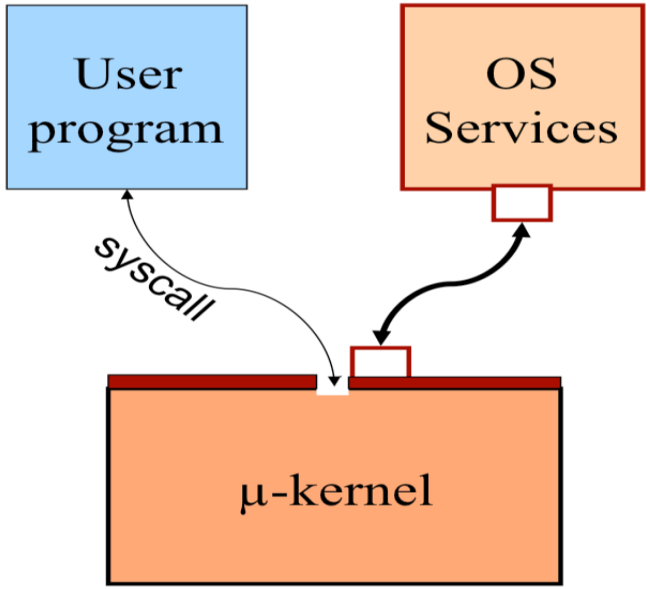


MULTICS

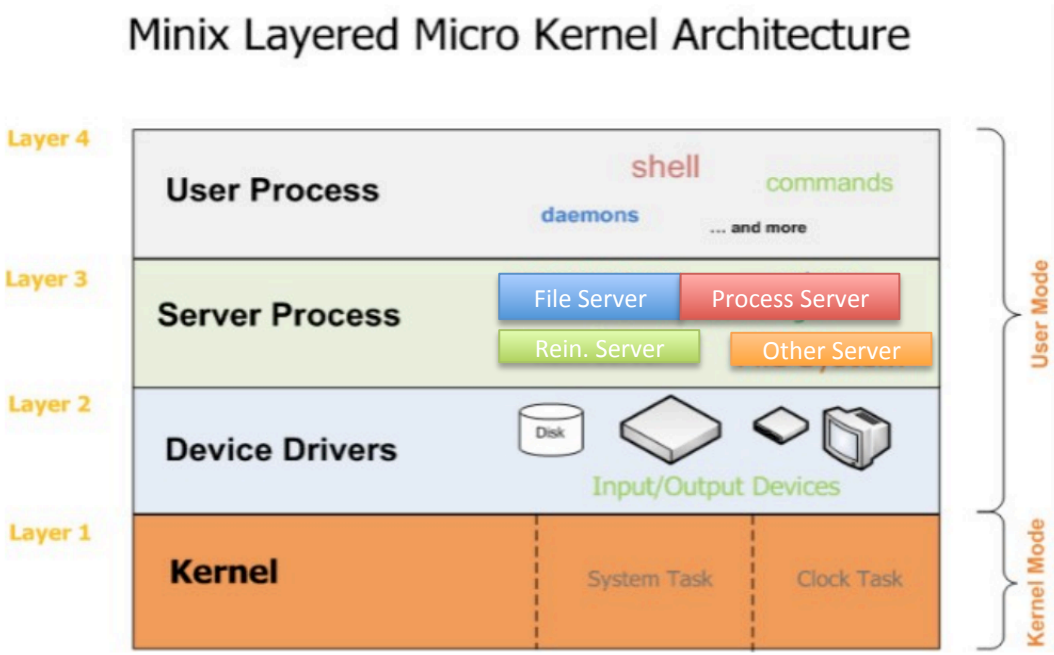
- Example: THE, MULTICS
- Advantages:
  - Layered Abstraction
- Disadvantages:
  - All the layers are linked together into a single executable program.

Microkernel

- The basic idea behind the microkernel design is to achieve high reliability by splitting the operating system up into small, well-defined modules,
- Only the microkernel—runs in kernel mode and the rest run as relatively powerless ordinary user processes



Microkernel



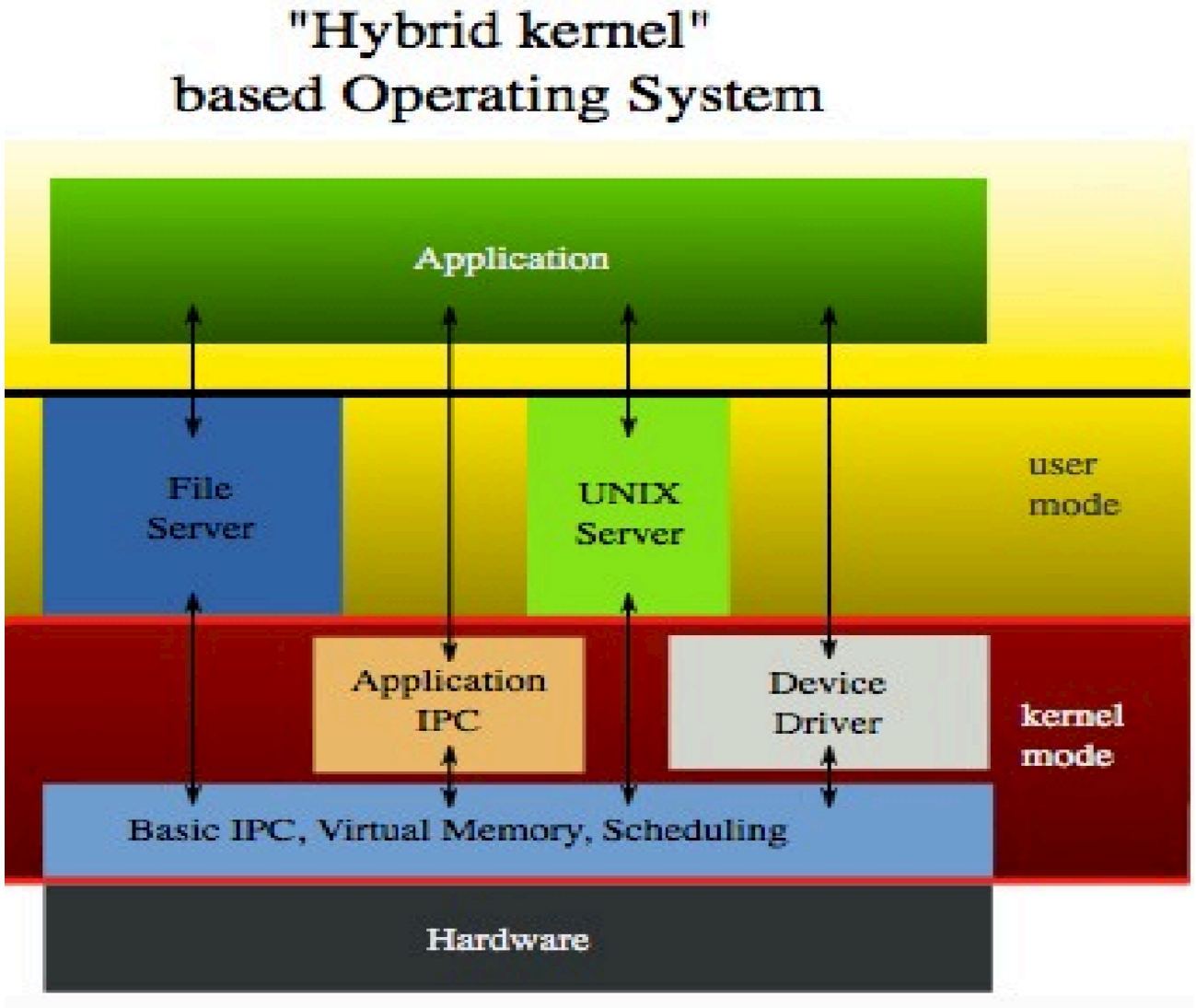
Kernel Mode:

1. System Task: - Catch interrupt
   - Switch process
   - Schedule process
   - Manage inter process communication
   - Provide other 35 system call service
2. Clock Task: - Contain driver for processor clock in order to assist scheduler.

User Mode:

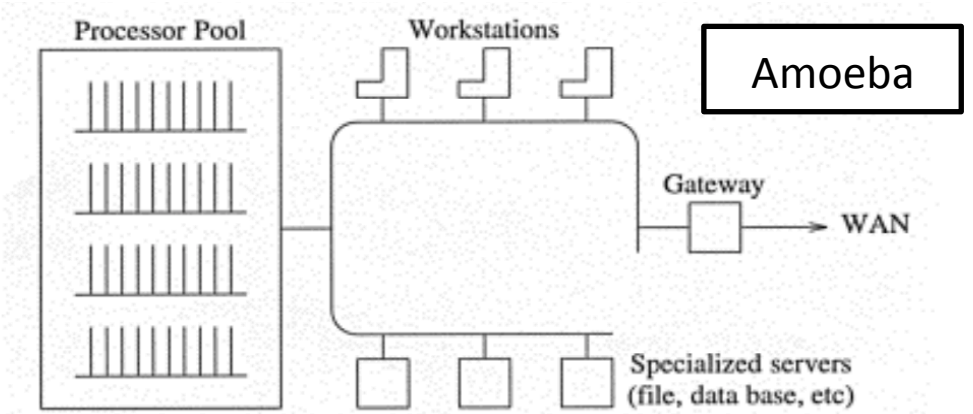
- Device Driver: Do not have access to I/O port. Only packets the value to be written at a port.
- Servers :
  - File servers: manage file system
  - Process server: creates, destroys and manages process.
  - Reincarnation server: Checks if other server and driver are functioning correctly.

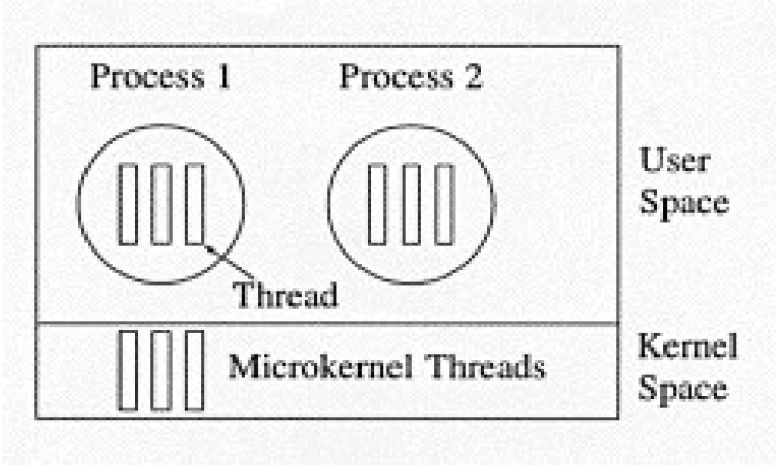
Hybrid


"Hybrid kernel"
based Operating System
Application
File
Server
UNIX
Server
user
mode
Application
IPC
Device
Driver
kernel
mode
Basic IPC, Virtual Memory, Scheduling
Hardware

Distributed

- The Bullet Server
  - Used for file storage
- The Directory Server
  - Used for file naming
  - Maps from names to capabilities
- The Replication Server
  - Used for fault tolerence and performance
- The Run Server
  - Run server manages the processor pools
- The Boot Server
  - Ensures that servers are up and running
  - If it discovers that a server has crashed,
    - it attempts to restart it, otherwise selects another processor to provide the service





Distributed

The Amoeba System Architecture

Amoeba is a microkernel-based operating system. It offers multithreaded programs (shared memory) and a remote procedure call (RPC) mechanism for communication between threads, potentially across the network; even kernel-threads use this RPC mechanism for communication.

Four basic components:

1. Each user has a workstation running Windows X or Unix Emulation
2. Pool of processors which are dynamically allocated to users as required.
3. Specialized servers: file, directory, database, etc.
4. These components were connected to each other by a fast LAN, and to the wide area network by a gateway.

Virtual Machine

- Provide an interface that looks like independent hardware -
  - To multiple different Oss' running simultaneously on the same physical hardware.
  - Each OS believes that it has access to and control over its own CPU, RAM, I/O devices, hard drives, etc.

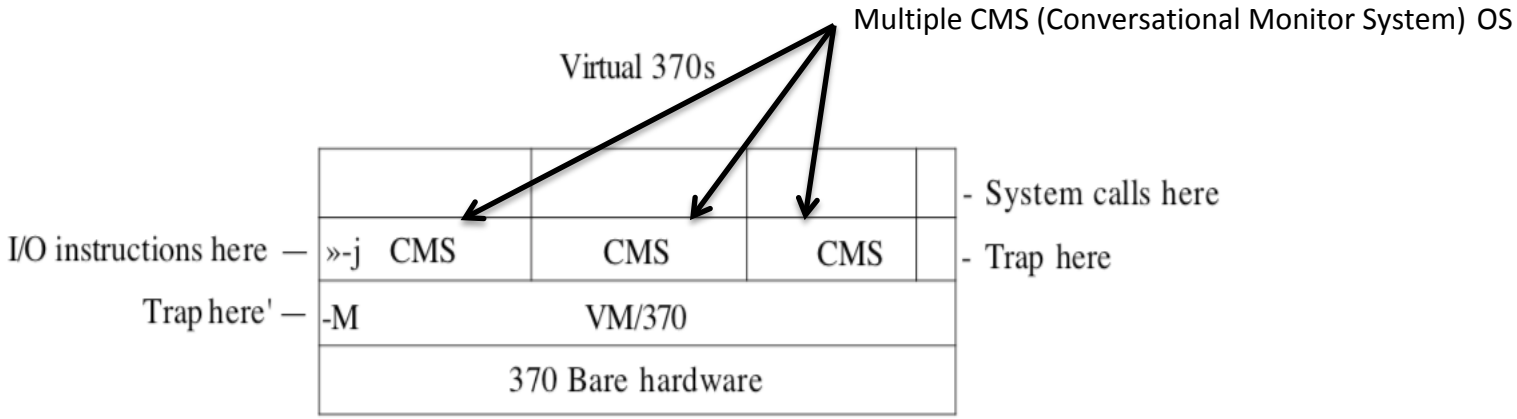
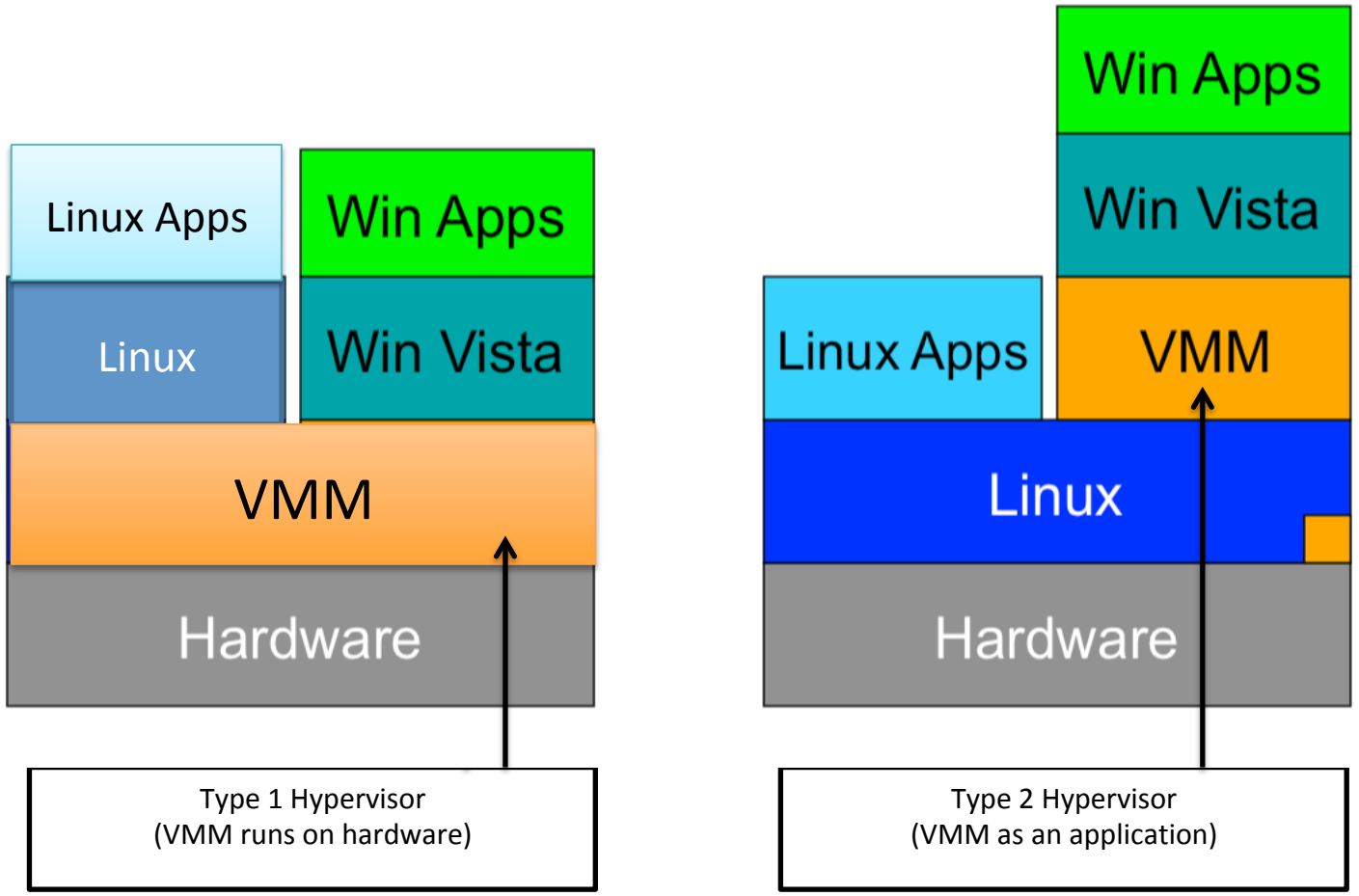


Figure 1-28. The structure of ViM/370 with CMS.

- Examples: IBM VM/370, Java VM, VMWare

Virtual Machine Organization



Reference

- https://www.cs.princeton.edu/
- https://www.inf.ed.ac.uk
- https://fenix.tecnico.ulisboa.pt
- http://fsd-amoeba.sourceforge.net/amoeba.html
- Book Ref:
    1. Modern Operating System − by Tanenbaum
    2. Operating System Concepts - by Abraham Silberschatz