به نام خدا

ارائه درس آزمایشگاه سیستم های عامل

موضوع: DISTRIBUTED OPERATING SYSTEMS

دانشجو: فرشید نوشی

پاییز ۱۴۰۰

TYPES OF OPERATING SYSTEMS

- **Batch Operating System**
- **Time-Sharing Operating Systems**
- **Distributed Operating Systems**
- **Network Operating Systems**
- **Real-time Operating Systems**

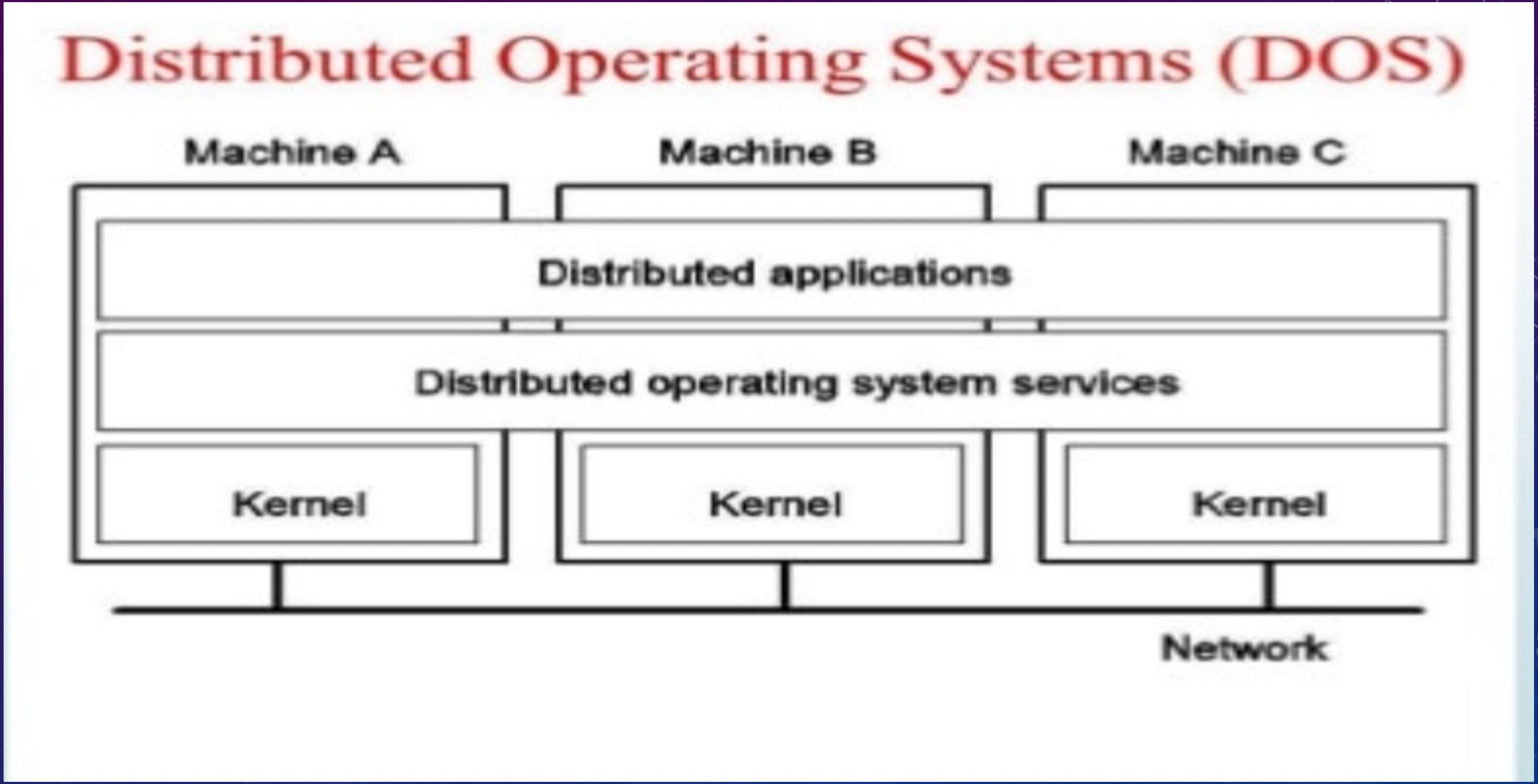
INTRODUCTION

These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace. Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as **loosely coupled systems** or distributed systems. These system's processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.

INTRUDUCTION

- Distributed Operating System is a model where distributed applications are running on multiple computers linked by communications.
- A Distributed operating system is an extension of the network operating system that supports higher levels of communication and integration of the machines on the network.
- This system looks to its users like an ordinary centralized operating system but runs on multiple, independent CPUs.
- Distributed systems use multiple central processors to serve multiple real time application and multiple users. Data processing jobs are distributed among the processors accordingly to which one can perform each job most efficiently.

Distributed Operating Systems (DOS)



Machine A

Machine B

Machine C

Distributed applications

Distributed operating system services

Kernel

Kernel

Kernel

Network

HISTORY

- Research and experimentation efforts began in earnest in 1970s and continued through 1990s, with focused interest peaking in the late 1980s. A number of distributed operating systems were introduced during this period; however, very few of these implementations achieved even modest commercial success.
- Fundamental and pioneering implementations of primitive distributed operating system component concepts date to the early 1950s. Some of these individual steps were not focused directly on distributed computing, and at the time, many may not have realized their important impact. These pioneering efforts laid important groundwork, and inspired continued research in areas related to distributed computing.
- In the mid-1970s, research produced important advances in distributed computing. These breakthroughs provided a solid, stable foundation for efforts that continued through the 1990s.

1950s- THE DYSEAC

- One of the first efforts was the DYSEAC, a general-purpose synchronous computer.
- This is one of the earliest examples of a computer with distributed control.

  The Dept. of the Army reports certified it reliable and that it passed all acceptance tests in April 1954.

  It was completed and delivered on time in May 1954.
- This was a “portable computer”, housed in a tractor-trailer, with 2 attendant vehicles and 6 tons of refrigeration capacity.

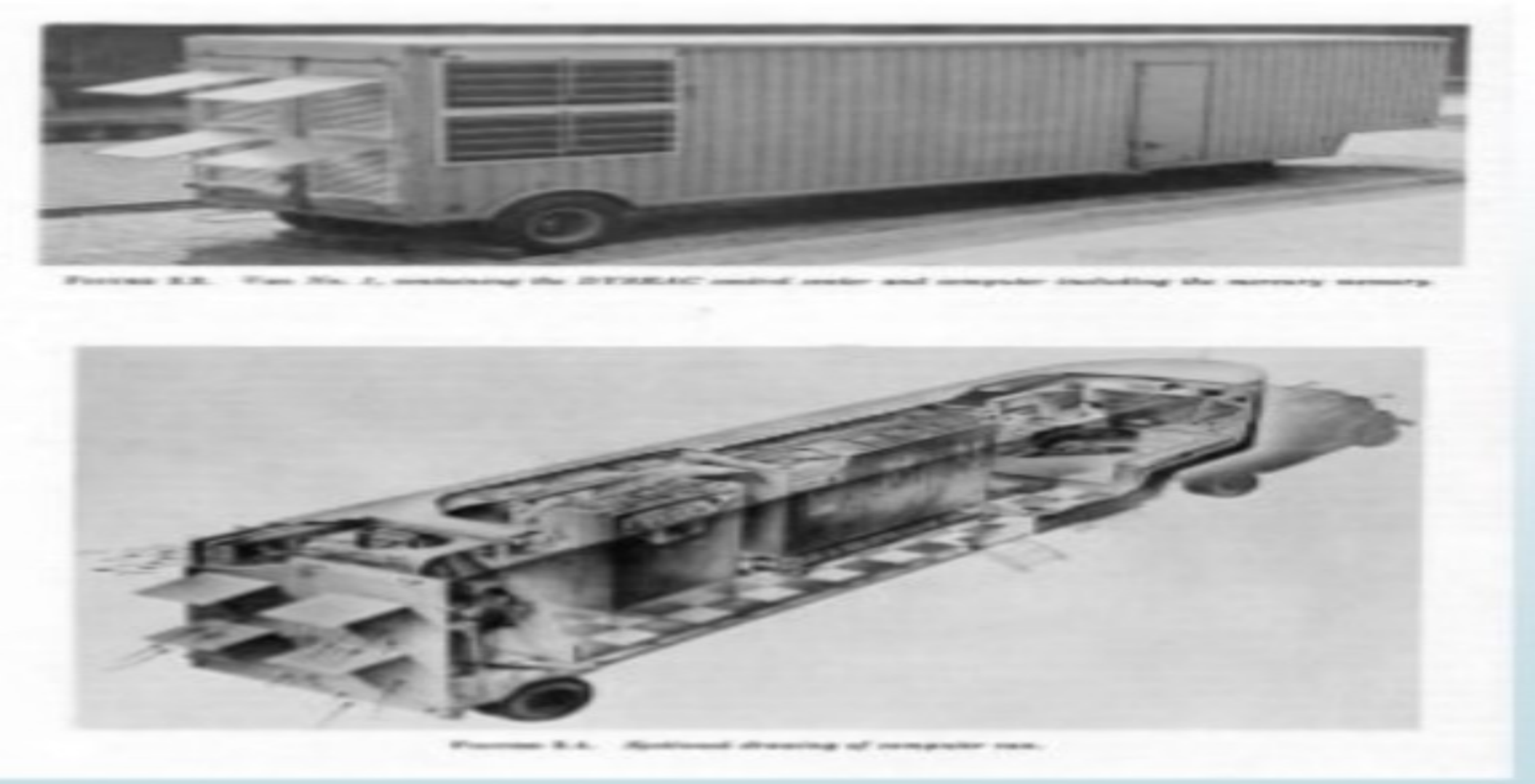


FIGURE 3.3. Van No. 1, containing the DYSEAC control center and computer including the mercury memory.

FIGURE 3.4. Phantom drawing of computer van.

LINCOLN-TX$_2$

- Described as an experimental input-output system, the Lincoln TX-2 emphasized flexible, simultaneously operational I/O devices, i.e. multiprogramming. The design of the TX-2 was modular, supporting a high degree of modification and expansion.
- The system employed The Multiple-Sequence Programming Techniques. This technique allowed multiple program counters to each associate with one of 32 possible sequences of program code. These explicitly prioritized sequences could be interleaved and executed concurrently, affecting not only the computation in process, but also the control flow of sequences and switching of devices as well.
- Similar to DYSEAC the TX-2 separately programmed devices can operate simultaneously, increasing throughput. The full power of the central unit was available to any device. The TX-2 was another example of a system exhibiting distributed control, its central unit not having dedicated control.

DESIGN CONSIDERATION

TRANSPARENCY

- Transparency or single-system image refers to the ability of an application to treat the system on which it operates without regard to whether it is distributed and without regard to hardware or other implementation details. Many areas of a system can benefit from transparency, including access, location, performance, naming, and migration.
- The consideration of transparency directly affects decision making in every every aspect of design of a distributed operating system. Transparency can impose certain requirements and/or restrictions on other design considerations.

INTER-PROCESS COMMUNICATION

- Inter-Process communication is the implementation of general communication, process interaction, and dataflow between threads and/or processes both within a node, and between nodes in a distributed OS.
- The intra-node and inter-node communication requirement drive low-level IPC design, which is the typical approach to implementing communication functions that support transparency.
- In this sense, Inter process communication is the greatest underlying concept in the low-level design considerations of a distributed operating system.

PROCESS MANAGEMENT

- Process management provides policies and mechanisms for effective and efficient sharing of resources between distributed processes.
- These policies and mechanisms support operations involving the allocation and de-allocation of processes and ports to processors, as well as mechanisms to run, suspend, migrate, halt, or resume process execution.
- While these resources and operations can be either local or remote with respect to each other, the distributed OS maintains state and synchronization over all processes in the system.

RESOURCE MANAGEMENT

- Systems resources such as memory, files, devices, etc. Are distributed throughout a system, and at any given moment, any of these nodes may have light to idle workloads.
- Load sharing and load balancing require many policy-oriented decisions, ranging from finding idle CPUs, when to move, and which to move.
- Many algorithms exist to aid in these decisions; however, this calls for a second level of decision making policy in choosing the algorithm best suited for the scenario, and the conditions surrounding the scenario.

RELIABILITY

- Distributed OS can provide the necessary resources and services to achieve high levels of reliability, or the ability to prevent and/or recover from errors.
- Faults are physical or logical defects that can cause errors in the system.
- For a system to be reliable, it must somehow overcome the adverse effects of faults.
- The primary methods for dealing with faults include fault avoidance, fault tolerance, and fault detection and recovery.
- Fault avoidance covers proactive measures taken to minimize the occurrence of faults.

PERFORMANCE

- Many benchmarks metrics quantify performance; throughput, response time, job completions per unit time, system utilization, etc.
- With respect to a distributed OS, performance most often distills to a balance between process parallelism and IPC.
- Managing the task granularity of parallelism in a sensible relation to the messages required for support is extremely effective.
- Also, identifying when it is more beneficial to migrate to a process to its data, rather than copy the data, is effective as well.

FLEXIBILITY

- Flexibility in a distributed OS is enhanced through the modular and characteristics of the distributed OS, and by providing a richer set of higher-level services.
- The completeness and quality of the kernel/microkernel simplifies implantation of such services, and potentially enables service providers greater choice of providers for such services.

ADVANTAGES OF DISTRIBUTED OPERATING SYSTEMS

- Give more performance than single system
- If one pc in distributed system malfunction or corrupts then other node or pc will take care of
- More resources can be added easily
- Resources like printers can be shared on multiple pc’s

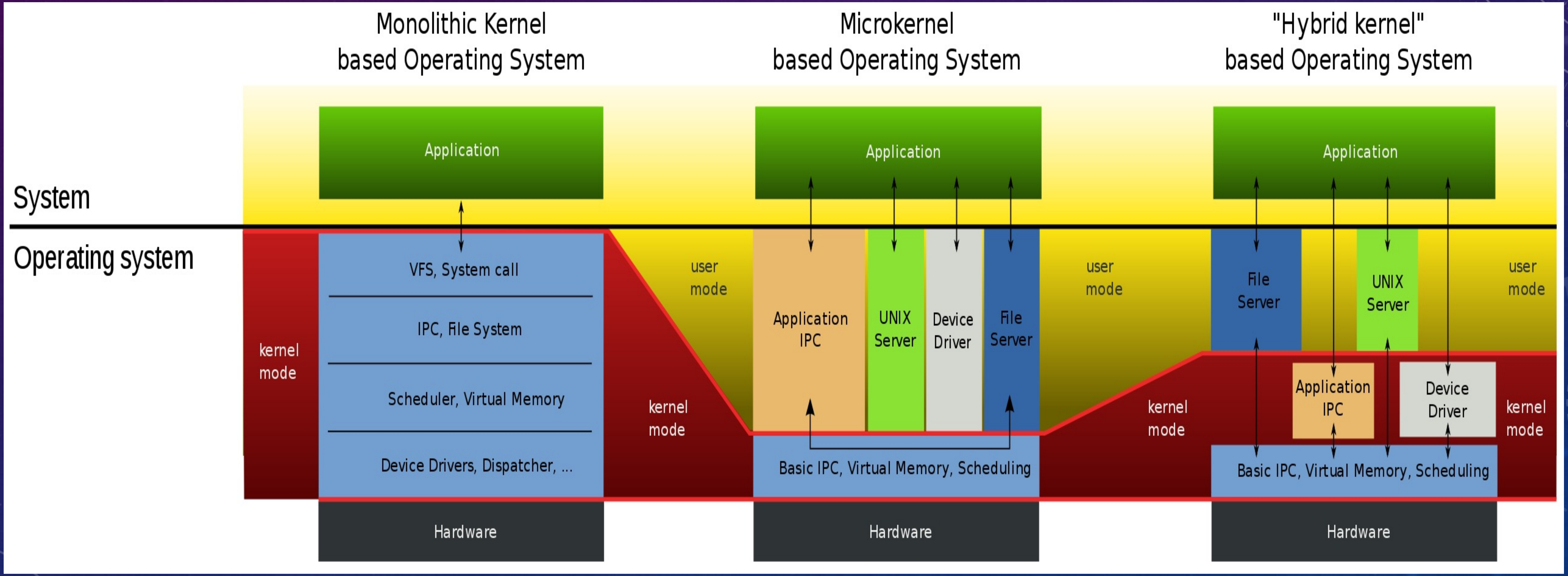
DISADVANTAGES OF DISTRIBUTED OPERATING SYSTEMS

- Security problems due to sharing
- Some messages can be lost in the network system
- Bandwidth is another problem if there is large data then all network wires to be replaces which tends to become expensive
- Overloading is another problem in distributed operating systems
- If there is a database connected on local system and many users accessing that database through remote or distributed way then performance become low
- The databases in network operating is difficult to administrate than single user system

EXAMPLES OF DISTRIBUTED OPERATING SYSTEM

- Windows server 2003
- Windows server 2008
- Windows server 2012
- Ubuntu
- Linux (Apache Server)

AN EXAMPLE


Monolithic Kernel
based Operating System
Microkernel
based Operating System
"Hybrid kernel"
based Operating System
System
Operating system
Application
VFS, System call
IPC, File System
Scheduler, Virtual Memory
Device Drivers, Dispatcher, ...
kernel mode
user mode
kernel mode
Hardware
Application
Application IPC
UNIX Server
Device Driver
File Server
user mode
Basic IPC, Virtual Memory, Scheduling
Hardware
kernel mode
Application
File Server
UNIX Server
user mode
Application IPC
Device Driver
kernel mode
Basic IPC, Virtual Memory, Scheduling
Hardware

THANK YOU