The background features a dark blue gradient with faint, light blue circular patterns. On the left side, there is a large circular scale with numerical markings from 150 to 260 in increments of 10. Several smaller circles with dashed outlines and arrows are scattered across the background, suggesting a technical or engineering theme.

EMBEDDED SYSTEMS DESIGN- UNIT IV TOPIC 1: OPERATING SYSTEM BASICS

BY

PRIYANKA YADLAPALLI

Operating System Basics

Operating System

Bridge between the application s/w and system resources

- Manages the resources
- Processes
- Time
- File System
- I/O System

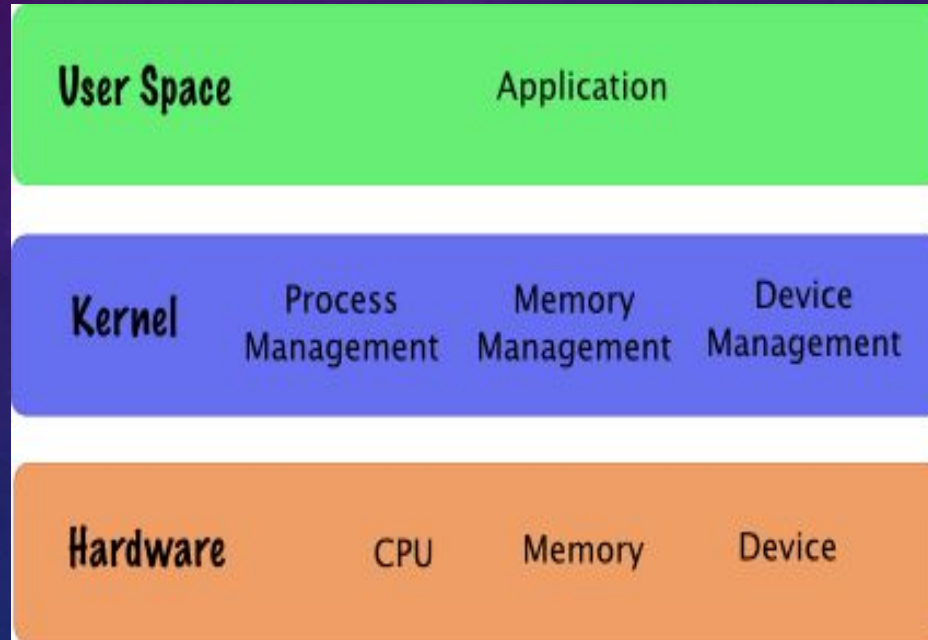


Fig.: Operating system architecture

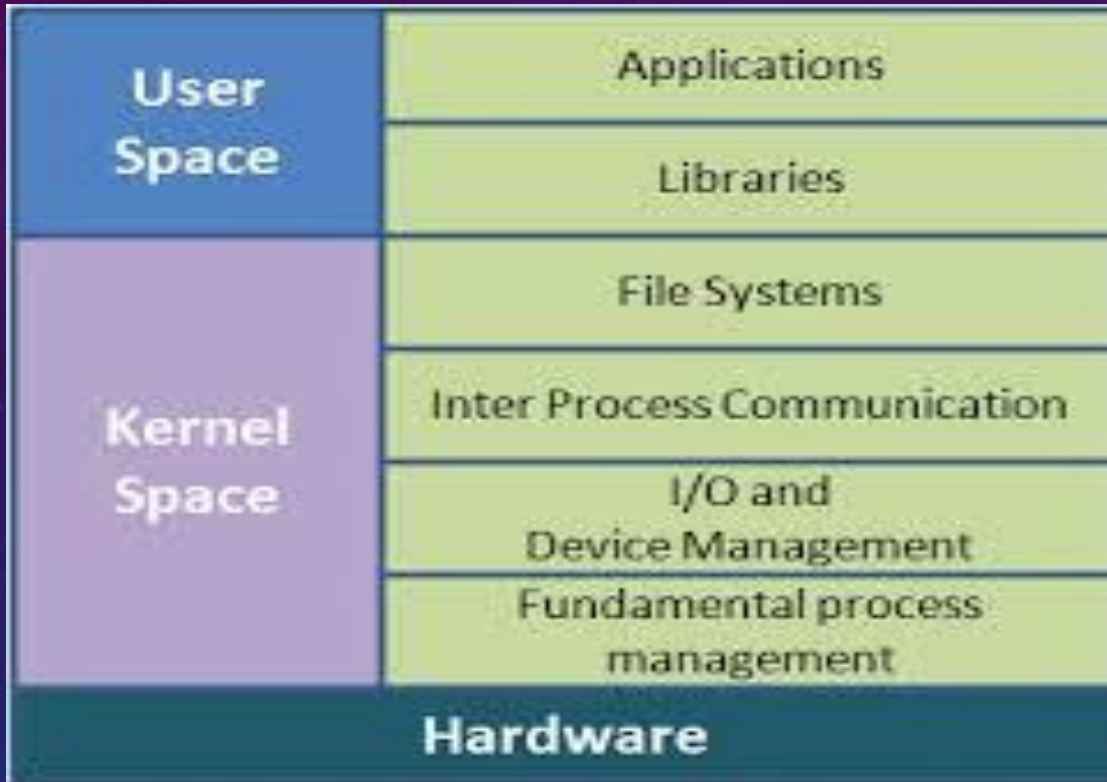


Fig.: Operating system architecture

Operating System Basics

The Kernel

The core of the OS

- ❑ Manages the system resources
- ❑ Communication among the system services
- ❑ Acts as the abstraction layer between system resource and user applications
- ❑ Contains system libraries and services

Operating System Basics

The Kernel contain different services for handling the following

- ❑ Process management
- ❑ Setting up the memory space
- ❑ Loading the process code into memory space
- ❑ Allocating system resources
- ❑ Scheduling and executing
- ❑ Setting up and managing Process Control Block (PCB)
- ❑ Inter Process Communication, and synchronisation
- ❑ Process termination/deletion

Primary Memory Management

- ✓ The Memory Management Unit (MMU) of the kernel
- ✓ Keeps track of which part of memory being used by which process is responsible for dynamic memory allocation

Operating System Basics

- File System Management service of kernel is responsible for
 - Creation, deletion, and modification of files, and directories
 - Saving of files in secondary storage memory
 - Automatic allocation of file space based on available free space
 - Naming convention
 - File system management operations are OS dependent.

Operating System Basics

I/O System(Device) Management

- ❑ Maintains a list of all the I/O devices
- ❑ Routes the I/O requests from user applications to appropriate I/O devices
- ❑ Provides Application Programming Interfaces (APIs)
- ❑ Updates the list of available devices dynamically when a new device is installed.
- ❑ Interacts with an I/O devices through a set of system calls which are implemented in a service called Device Driver.

- ❑ The Device Drivers are specific to the devices or class of devices.

- ❑ The service ‘Device Manager’ is responsible for
 - ❑ handling all I/O device related operations
 - ❑ Loading and unloading of device drivers
 - ❑ Exchanging information and control signals to and from the device.

Operating System Basics

Secondary Storage Management

The secondary storage management service of the kernel deals with

- Disk storage allocation
- Disk scheduling
- Free Disk space management

Protection Systems

This service of the kernel

- Implement security policies in multiuser supported OS one user can access other user's data without authority
- Some of the system resources can be restricted for some users

Operating System Basics

Kernel Space and User Space

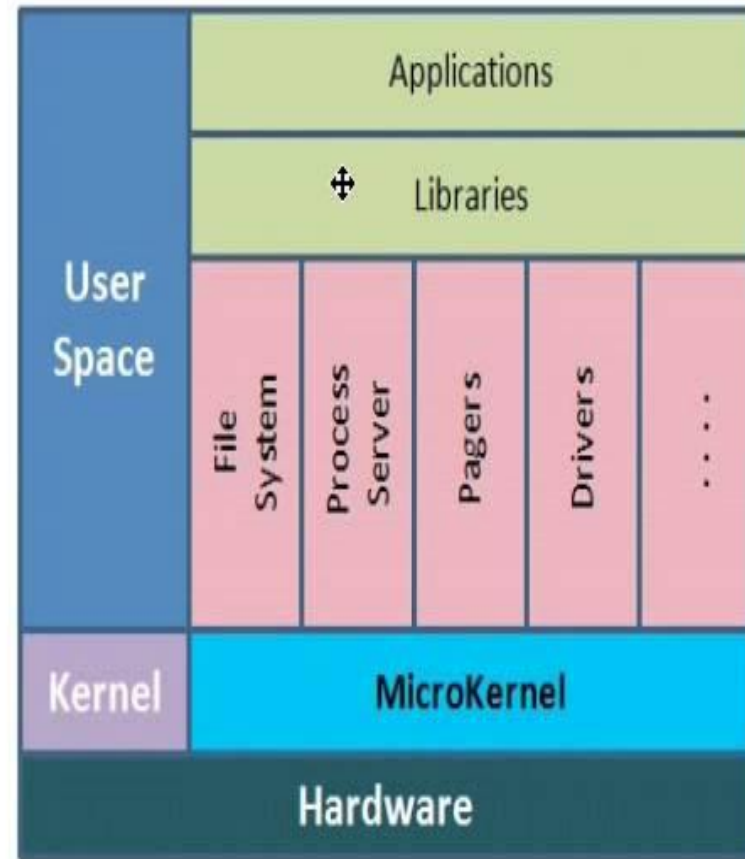
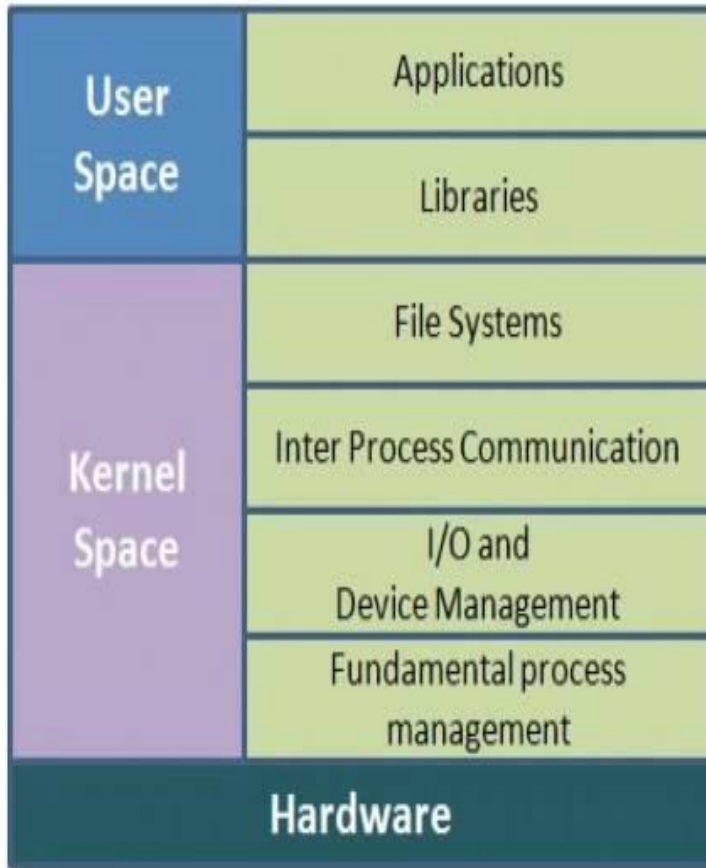
Kernel Space

- ❑ A contiguous memory area (OS dependent) of primary memory where kernel code is kept.
- ❑ Protected from unauthorised access by user program/applications

User Space

- ❑ Memory space where user programs are loaded and executed.
- ❑ Some Operating Systems do not segregate the kernel space and user space.
- ❑ Most of the operating systems keep the kernel in main memory and it is not swapped out into the secondary memory

Monolithic Kernel vs Microkernel



Operating System Basics

Monolithic Kernel and Microkernel

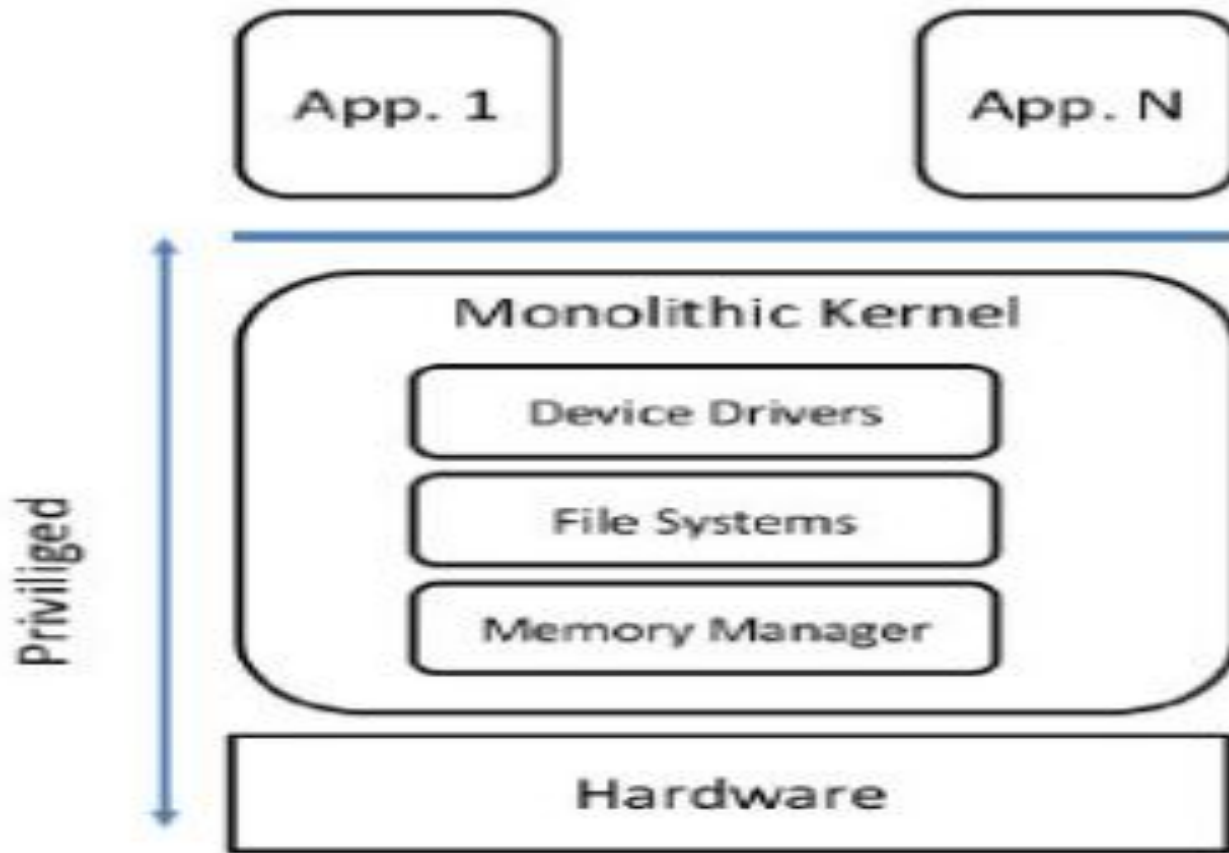
Monolithic Kernel

- All kernel services run in the kernel space
- All kernel modules run within the same memory space under a single thread.

Major drawback

- Any error/failure in any of the modules leads to the crashing of the entire kernel.

Examples: Linux, Solaris, MS-DOS



A: Monolithic kernel design

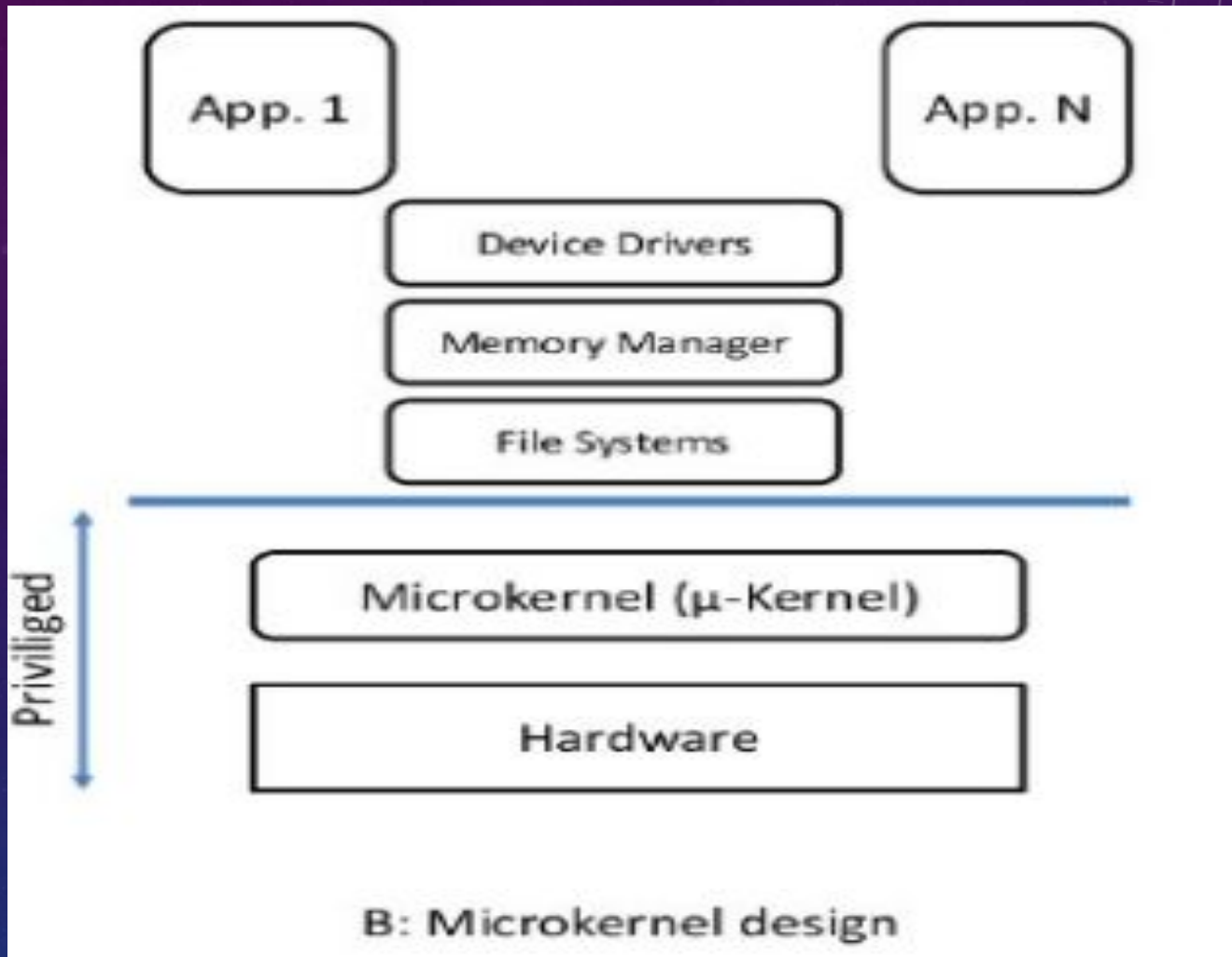
Micro Kernel

- Incorporates only the essential set of services
- Other services are implemented in the programs called as servers run in user space
- Highly modular design

Examples: Mach, QNX, Minix 3

Advantages

- **Robustness:** if an error occur in any of the services running as server, then that service (server) can be reconfigured, and re-started without re-starting the whole OS.
- **Configurability:** any service runs as server application can be modified without restarting the entire OS. This makes the system dynamically configurable.



Basis for Comparison	Microkernel	Monolithic Kernel
Size	Microkernel is smaller in size	It is larger than microkernel
Execution	Slow Execution	Fast Execution
Extendible	It is easily extendible	It is hard to extend
Security	If a service crashes, it does effects on working on the microkernel	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel more code is required	To write a monolithic kernel less code is required
Example	QNX, Symbian, L4Linux etc.	Linux, BSDs (FreeBSD, OpenBSD, NetBSD) etc.