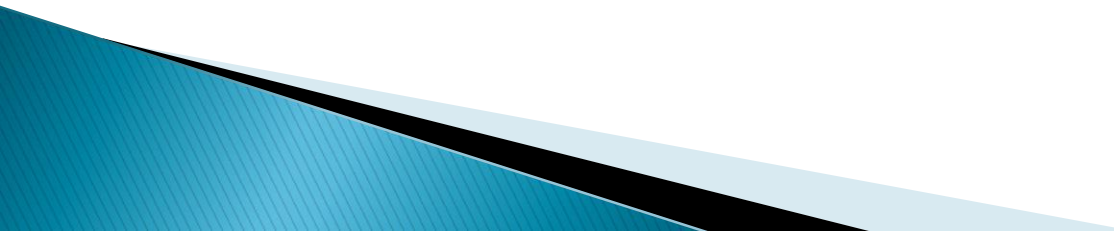


Free and Open Source Software (FOS)–21527

Presented By:
M. P. Wakchaure

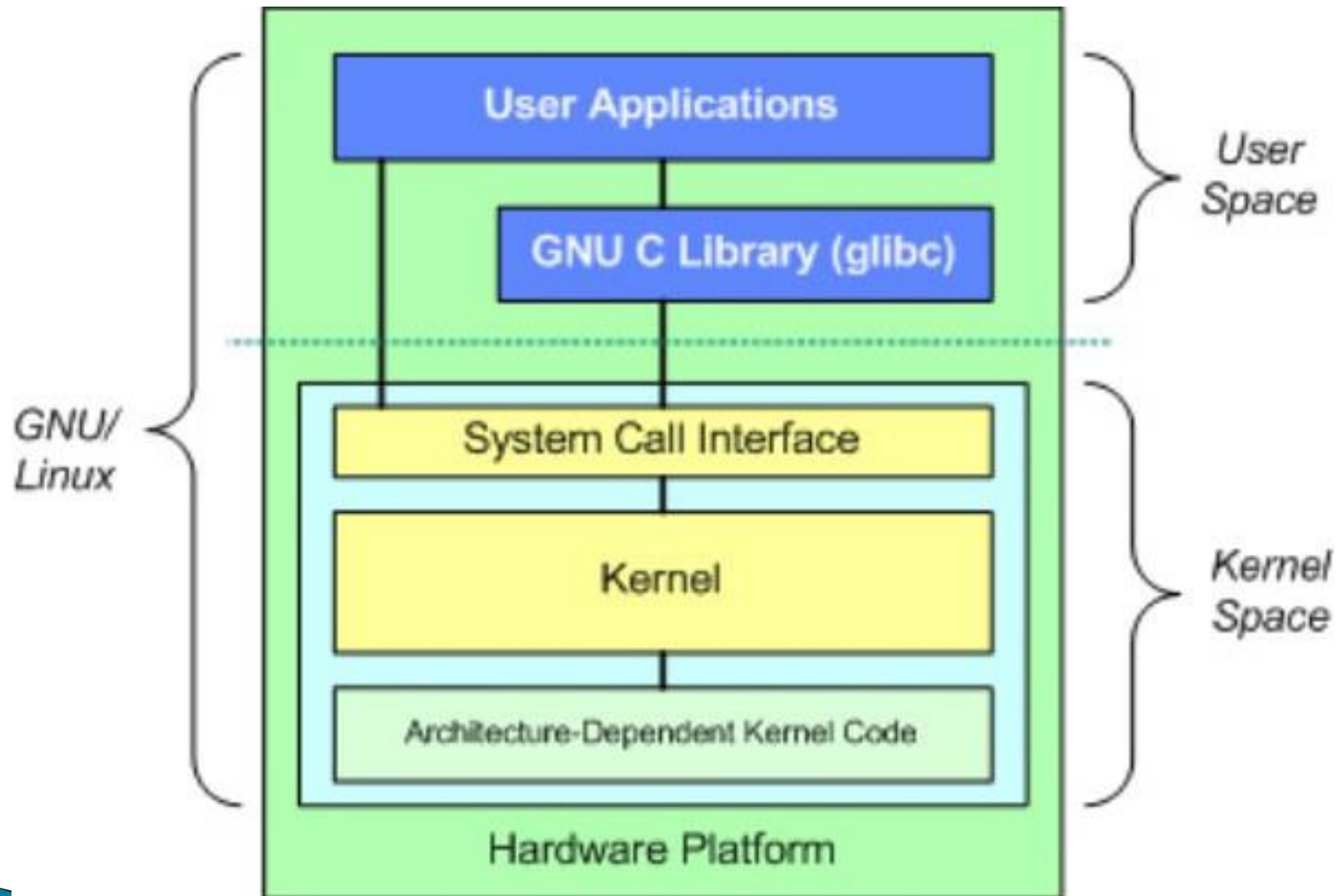
What is Linux?

- ▶ Linux is a Unix clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net.
 - ▶ Unix is a multitasking, multi-user computer operating system originally developed in 1969 by a group of AT&T employees at Bell Labs.
 - ▶ Linux and Unix strive to be POSIX compliant.
 - ▶ 64% of the world's servers run some variant of Unix or Linux. The Android phone and the Kindle run Linux.
- 

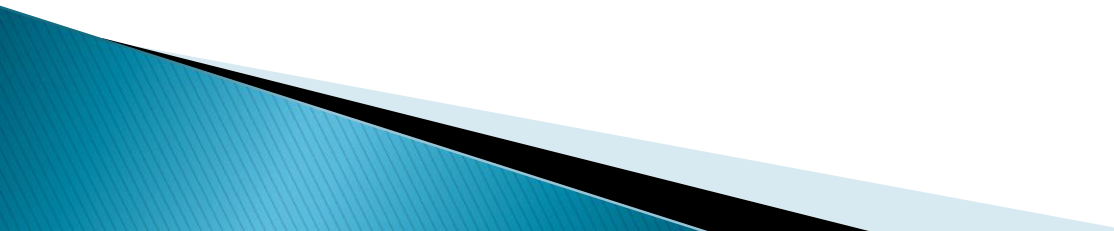
Components of Linux System

Linux Operating System has primarily three components

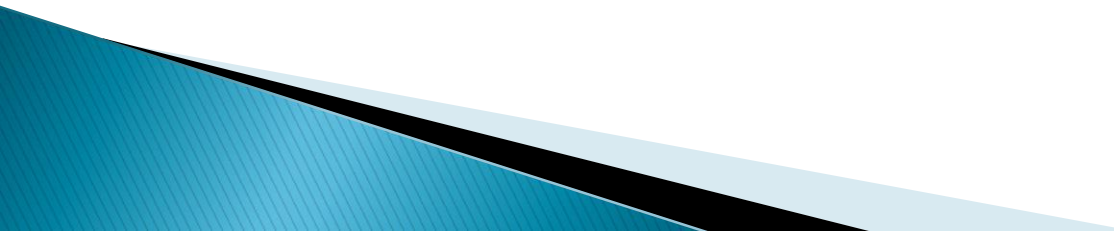
- ▶ Kernel
- ▶ System Library
- ▶ System Utility



Kernel

- ▶ Kernel is the core part of Linux.
 - ▶ It is responsible for all major activities of this operating system.
 - ▶ It consists of various modules and it interacts directly with the underlying hardware.
 - ▶ Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- 

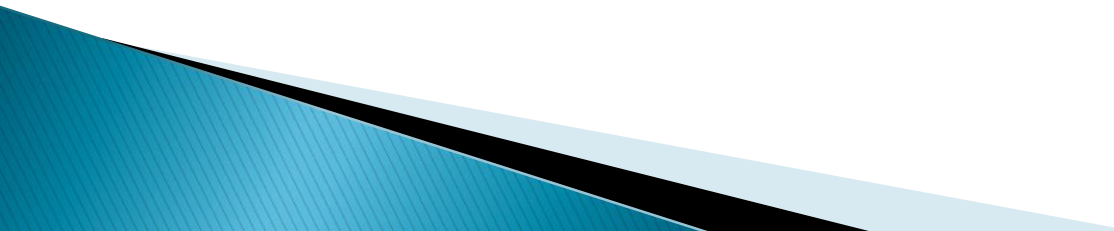
System Library

- ▶ System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features.
 - ▶ These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
- 

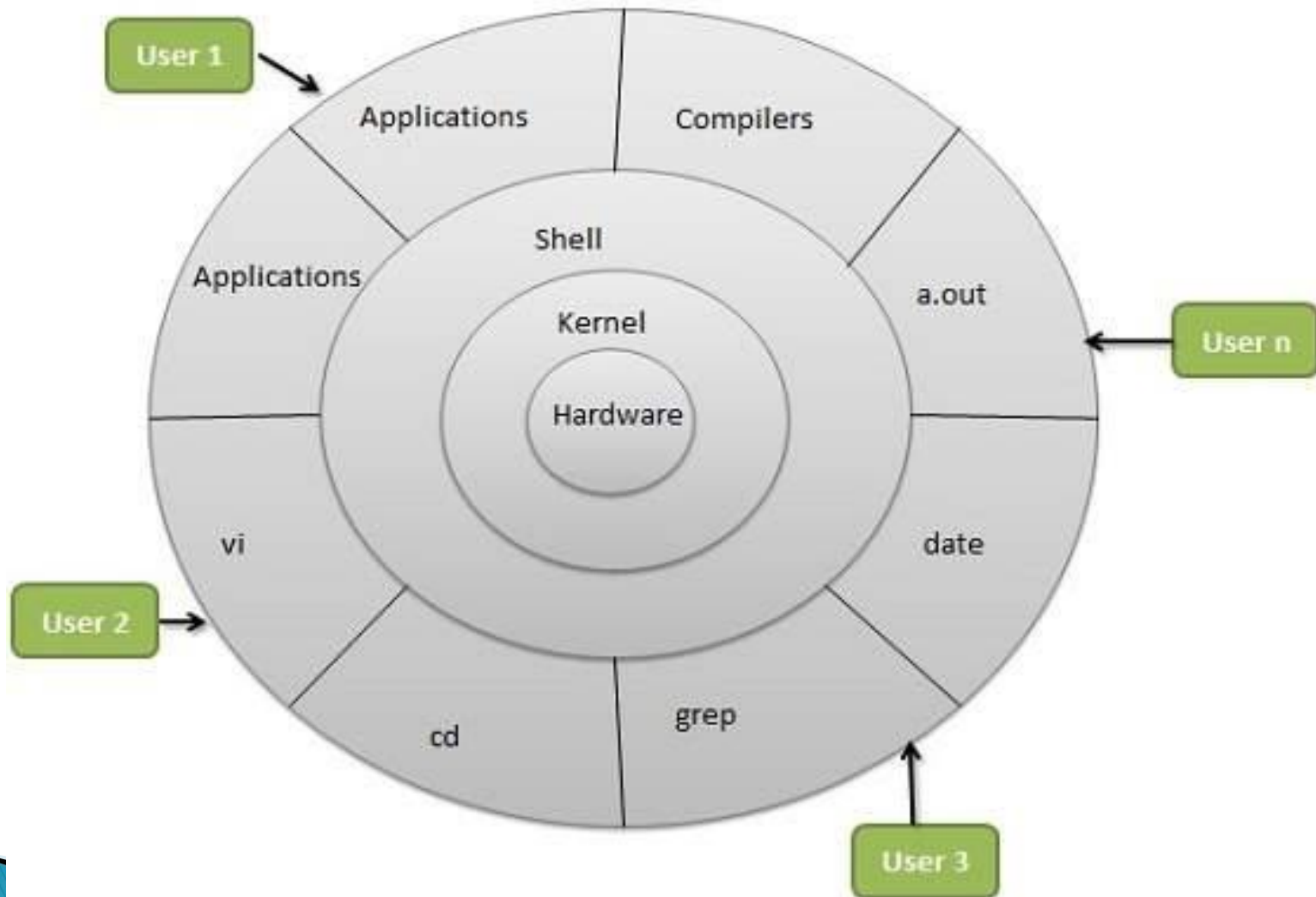
System Utility

- ▶ System Utility programs are responsible to do specialized, individual level tasks.

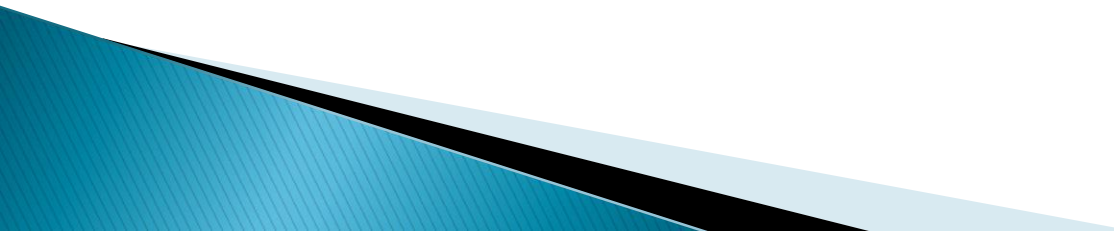
Basic Features

- ▶ Portable
 - ▶ Open Source
 - ▶ Multi-User
 - ▶ Multiprogramming
 - ▶ Hierarchical File System
 - ▶ Shell
 - ▶ Security
- 

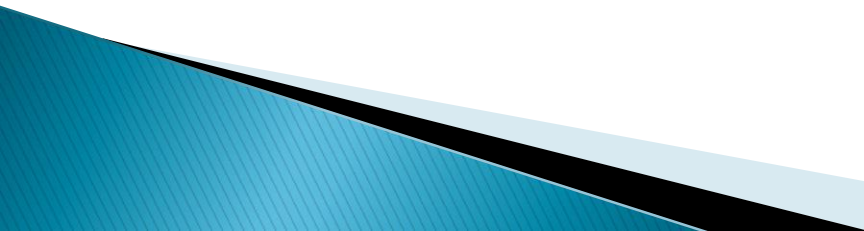
Architecture



The architecture of a Linux System consists of the following layers –

- ▶ **Hardware layer** – Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
 - ▶ **Kernel** – It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
 - ▶ **Shell** – An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
 - ▶ **Utilities** – Utility programs that provide the user most of the functionalities of an operating systems.
- 

Difference Between Kernel mode and User mode:

- ▶ Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer.
 - ▶ This code represents a single process, executes in single address space and do not require any context switch and hence is very efficient and fast.
 - ▶ Kernel runs each processes and provides system services to processes, provides protected access to hardwares to processes.
- 

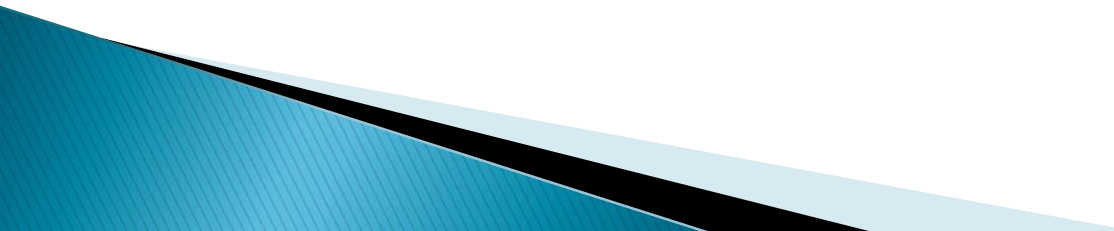
Support code which is not required to run in kernel mode is in System Library.

User programs and other system programs work in User Mode which has no access to system hardware and kernel code.

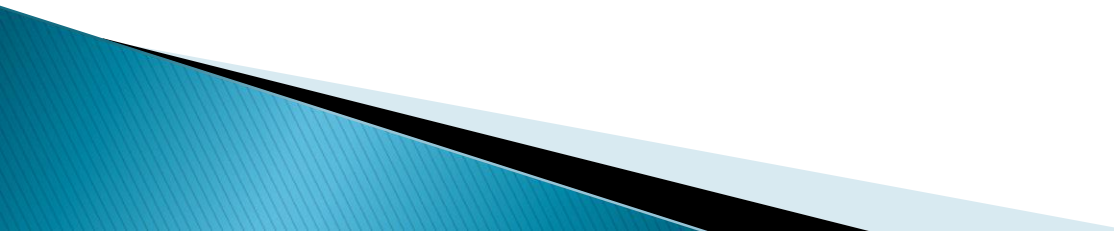
User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.



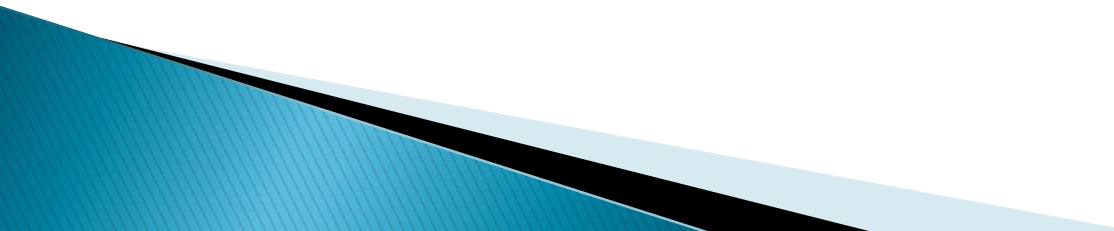
Application

- ▶ Dell Inspiron Mini 9 and 12
 - ▶ Garmin Nuvi 860, 880, and 5000
 - ▶ Google Android Dev Phone 1
 - ▶ HP Mini 1000
 - ▶ Lenovo IdeaPad S9
 - ▶ Motorola MotoRokr EM35 Phone
 - ▶ One Laptop Per Child XO2
 - ▶ Sony Bravia Television
 - ▶ Sony Reader
- 

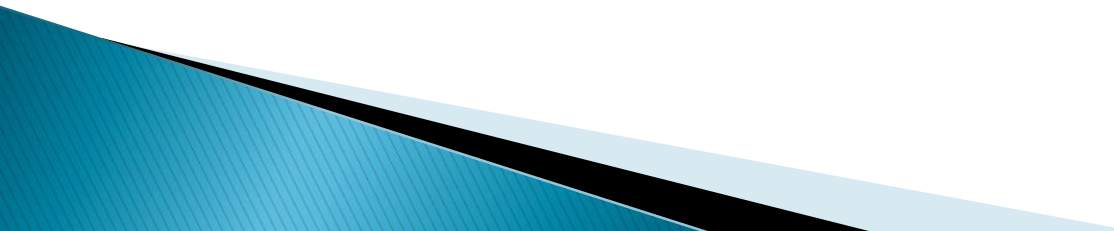
Advantages

- ▶ Linux is free and can be downloaded from the internet; there is no hidden cost for registration, updates, or anything.
 - ▶ Linux is flexible, i.e. **Linux can be installed** on any hardware; if a user is not sure what OS on his/her machine can be installed, he can go with Linux.
 - ▶ Linux is developed so that it can run all the time without rebooting, and because of this feature, many applications can be scheduled at calm hours.
 - ▶ The security model implemented for Linux is based on Unix; it is very secure against the internet and other attacks as well.
 - ▶ Linux can be customized as per requirement, and bug fixes are very fast as it is open source, and we can find many people available on the internet for solutions.
- 

Disadvantages

- ▶ Linux is licensed under GNU Public License (GPL), which suggests anyone can change and distribute a changed version. So it is a bit confusing to find a suitable for our need version.
 - ▶ Linux is not very user-friendly, and it can be a bit confusing for beginners.
- 

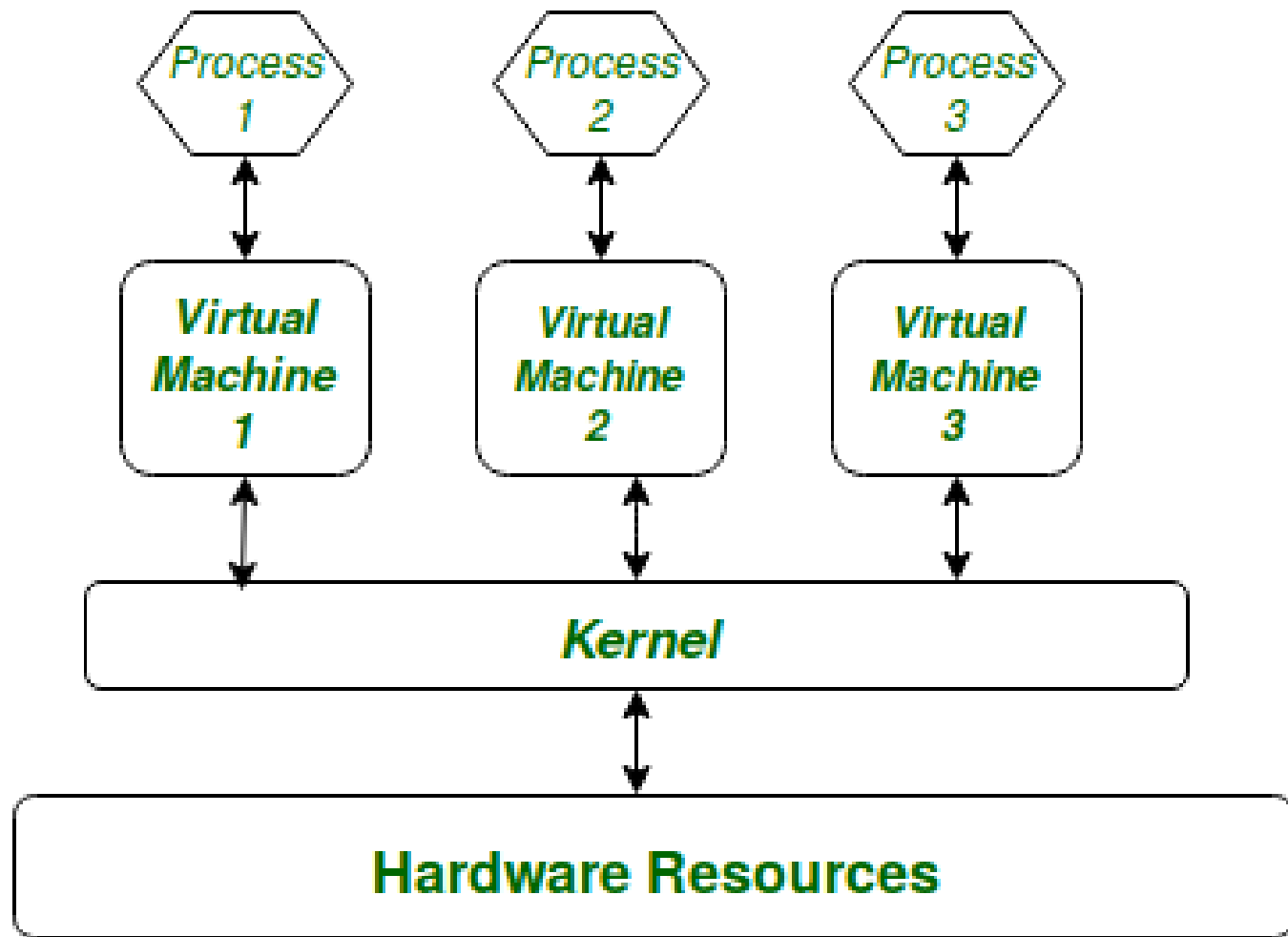
Kernel

- ▶ The kernel virtualizes the computer's common hardware resources to provide each process with its own virtual resources.
 - ▶ This makes the process seem as if it is the sole process running on the machine.
 - ▶ The kernel is also responsible for preventing and mitigating conflicts between different processes.
- 



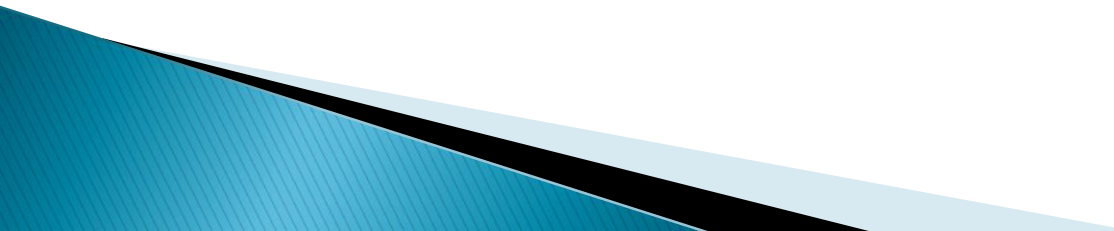
Design

- ◉ The Linux kernel has a monolithic design
- ◉ The other approach is the microkernel design
- ◉ Both have their upsides and downsides
 - > Monolithic kernels
 - Easier to build and design
 - Generally faster
 - More recompiles
 - Less object oriented
 - > Micro kernels
 - Considered safer
 - Easier to develop drivers for
 - Only recompile for upgrades
 - Generally slower
 - Much harder to build and design



Per-Process Hardware Virtualization

The Core Subsystems of the Linux Kernel are as follows:

- ▶ The Process Scheduler
 - ▶ The Memory Management Unit (MMU)
 - ▶ The Virtual File System (VFS)
 - ▶ The Networking Unit
 - ▶ Inter-Process Communication Unit
- 

The Linux Kernel Subsystems

*Process
Scheduler*

*The Memory
Management
Unit (MMU)*

*Virtual File
System (VFS)*

*The
Networking
Subsystem*


*The Inter-
Process
Communication
Unit*

The Linux Kernel

The Process Scheduler: This kernel subsystem is responsible for fairly distributing the CPU time among all the processes running on the system simultaneously.

The Memory Management Unit: This kernel sub-unit is responsible for proper distribution of the memory resources among the various processes running on the system. The MMU does more than just simply provide separate virtual address spaces for each of the processes.

The Virtual File System: This subsystem is responsible for providing a unified interface to access stored data across different filesystems and physical storage media.



What the kernel does

The kernel has 4 jobs:

Memory management: Keep track of how much memory is used to store what, and where

Process management: Determine which processes can use the central processing unit (CPU), when, and for how long

Device drivers: Act as mediator/interpreter between the hardware and processes

System calls and security: Receive requests for service from the processes



Reference:

- ▶ <https://www.linux-india.org/components-of-linux-system/>
- ▶ https://www.tutorialspoint.com/operating_system/os_linux.htm#