



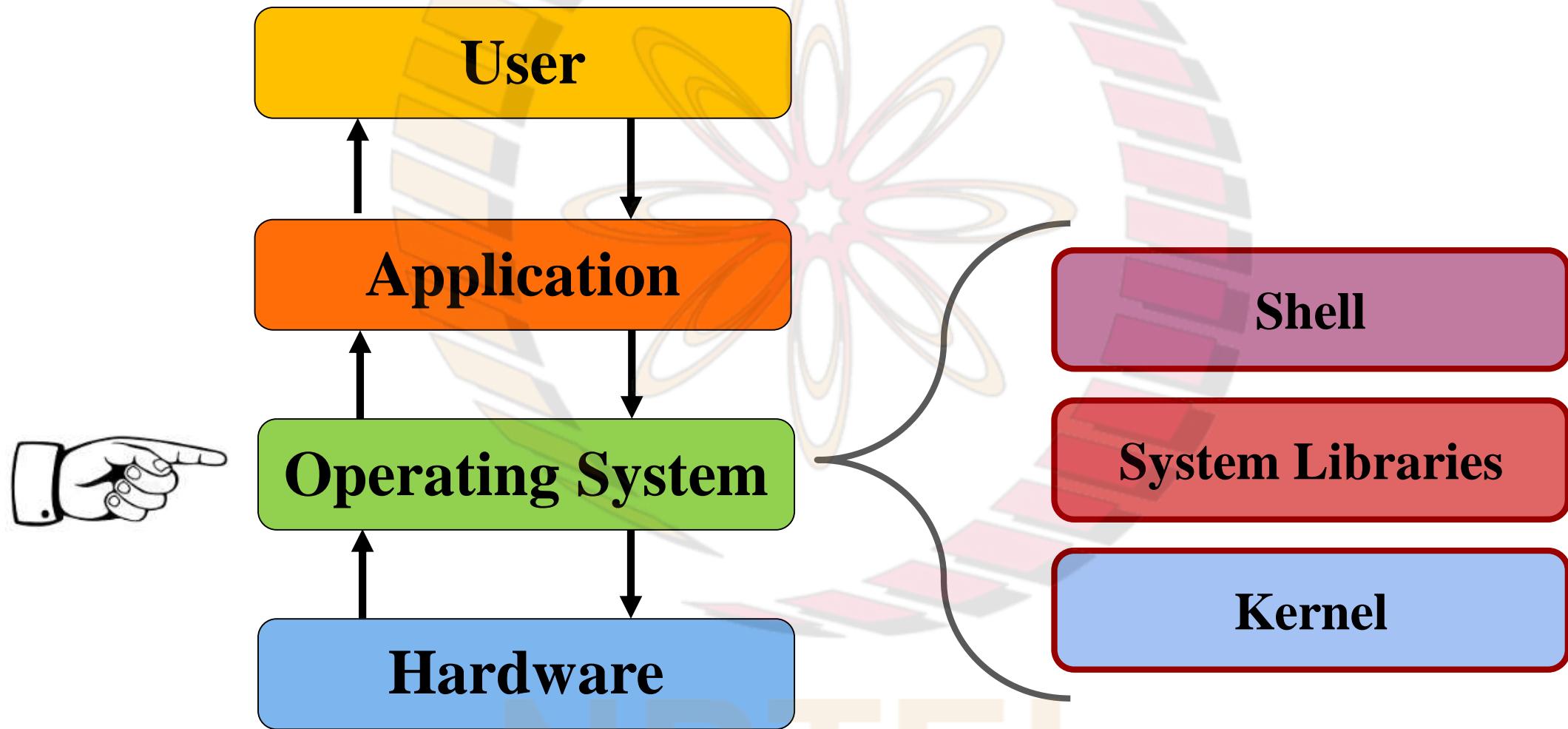
# **Introduction to Linux Operating System**

**NPTEL**

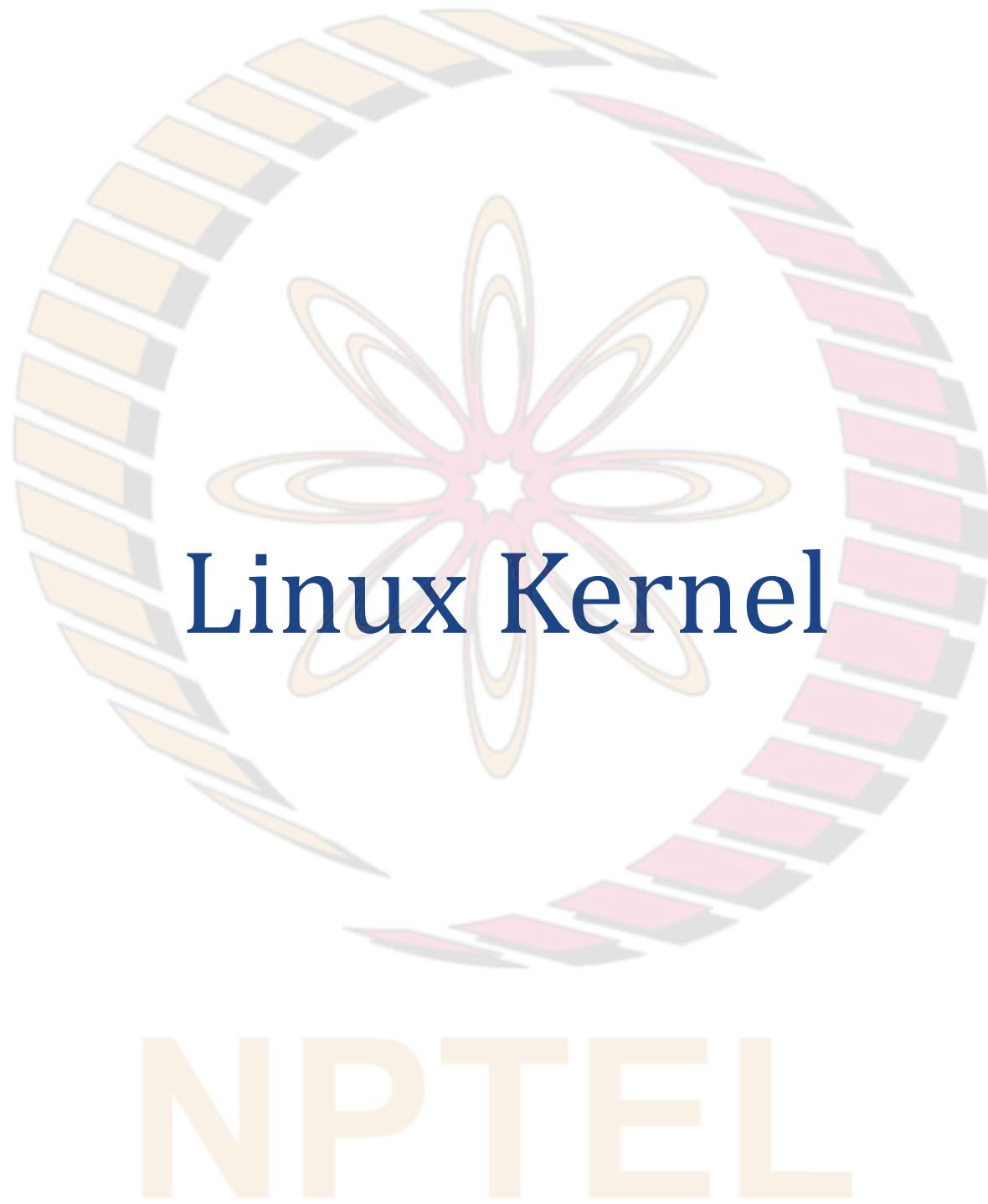
# Contents

- Linux Kernel
- Linux Kernel Sub-components
  - Process Scheduler
  - Memory Management Unit (MMU)
  - Virtual File System (VFS)
  - Network Sub-System
  - Inter Process Communication Unit
- Linux Library files and Directory Structure
- Shell
  - Types of Shells
- Linux Commands and It's Structure

# Recap..



NPTEL



# Linux: Kernel



“The kernel provides a platform for programs and various services to run on top of it”

NPTEL

# Linux: Kernel



“The kernel provides a platform for programs and various services to run on top of it”

- It acts as a bridge between software applications and the underlying hardware of a computer system

# Linux: Kernel



“The kernel provides a platform for programs and various services to run on top of it”

- It acts as a bridge between software applications and the underlying hardware of a computer system
- It is responsible for managing ***system resources***, handling ***hardware communication***, and ensuring smooth execution of processes.

# Linux: Kernel



“The kernel provides a platform for programs and various services to run on top of it”

- It acts as a bridge between software applications and the underlying hardware of a computer system
- It is responsible for managing ***system resources***, handling ***hardware communication***, and ensuring smooth execution of processes.
- The Linux kernel follows a ***monolithic architecture***, meaning that it runs as a single large process in a privileged mode.

# Linux: Kernel



“The kernel provides a platform for programs and various services to run on top of it”

- It acts as a bridge between software applications and the underlying hardware of a computer system
- It is responsible for managing ***system resources***, handling ***hardware communication***, and ensuring smooth execution of processes.
- The Linux kernel follows a ***monolithic architecture***, meaning that it runs as a single large process in a privileged mode.
- supports ***loadable kernel modules (LKMs)***, allowing dynamic addition or removal of features

# Linux: Kernel

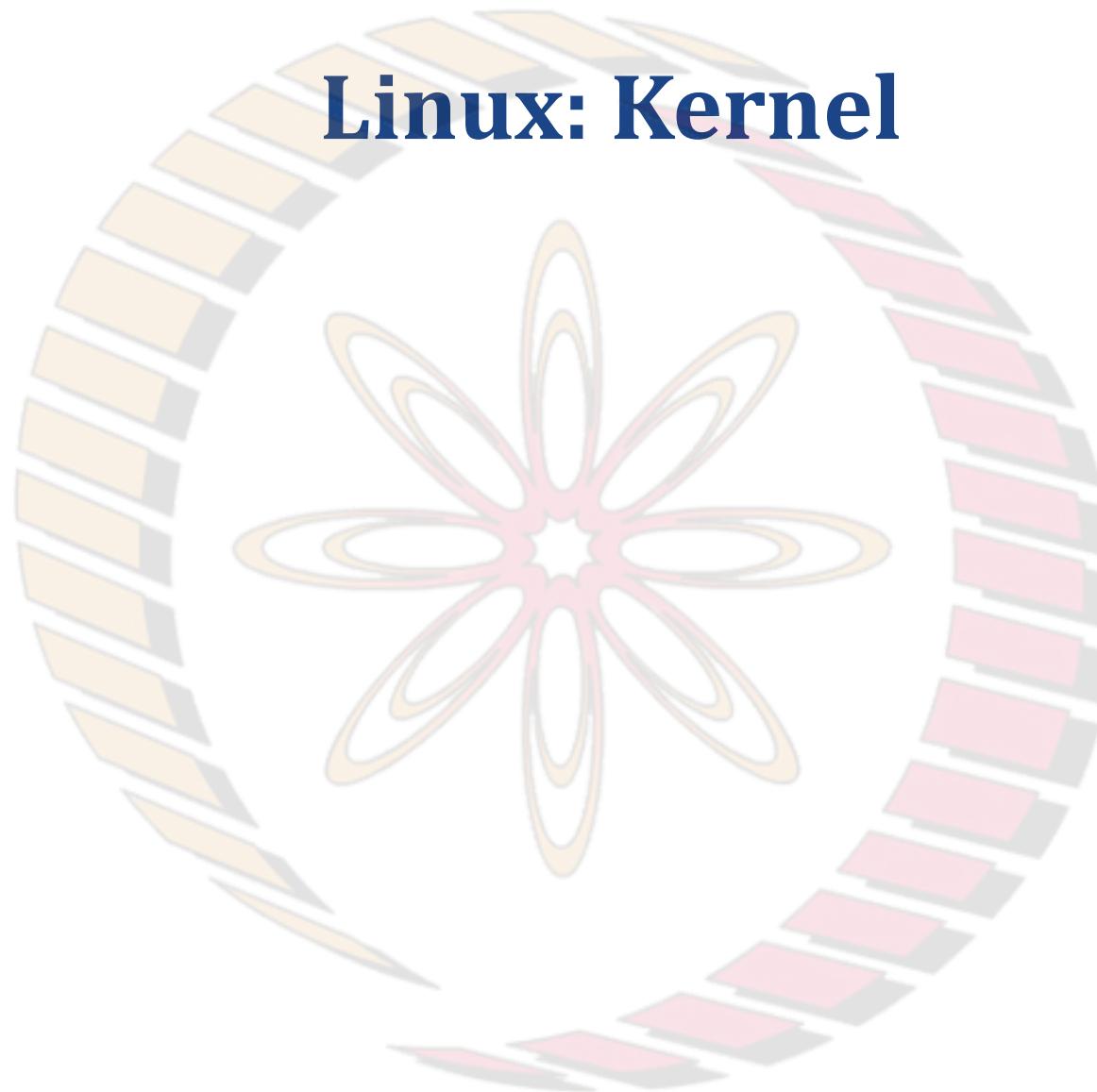
Kernel

“The kernel ***virtualizes*** the computer’s common hardware resources to provide each process with its own virtual resources”

NPTEL

# Linux: Kernel

Kernel



NPTEL

# Linux: Kernel

Kernel

Process-1

Process-2

Process-3

NPTEL

# Linux: Kernel

Kernel

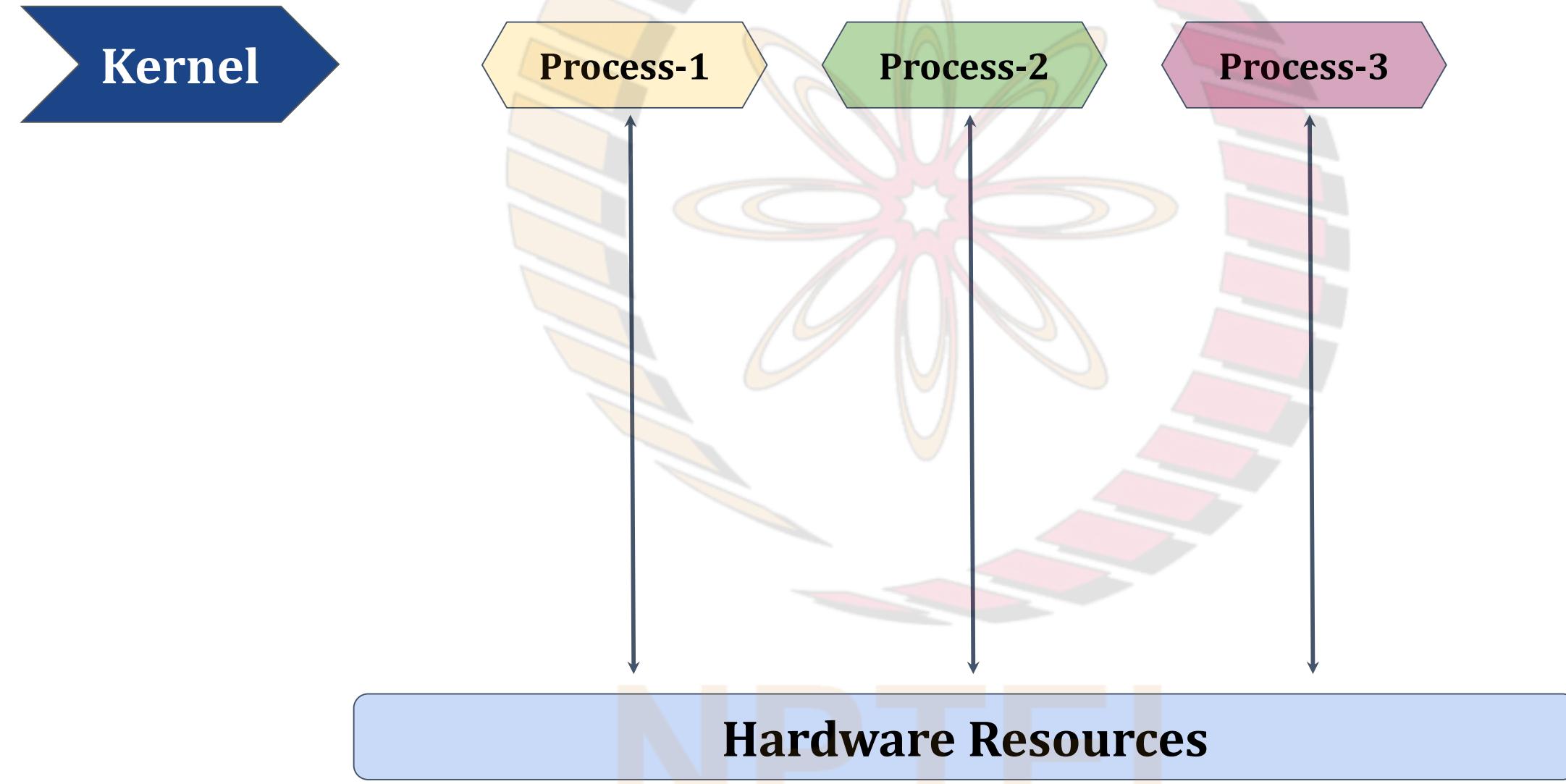
Process-1

Process-2

Process-3

Hardware Resources

# Linux: Kernel



# Linux: Kernel

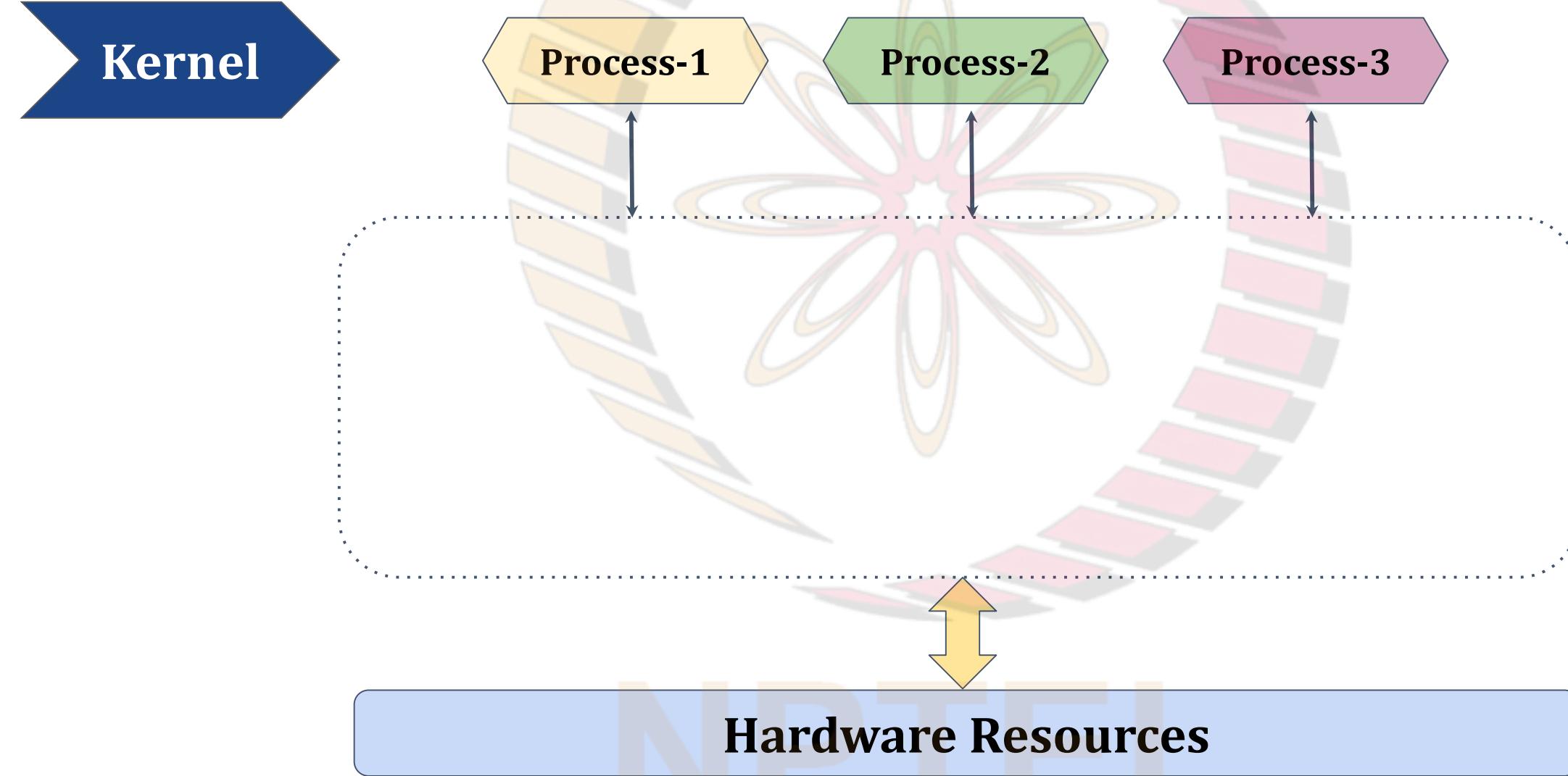
Kernel



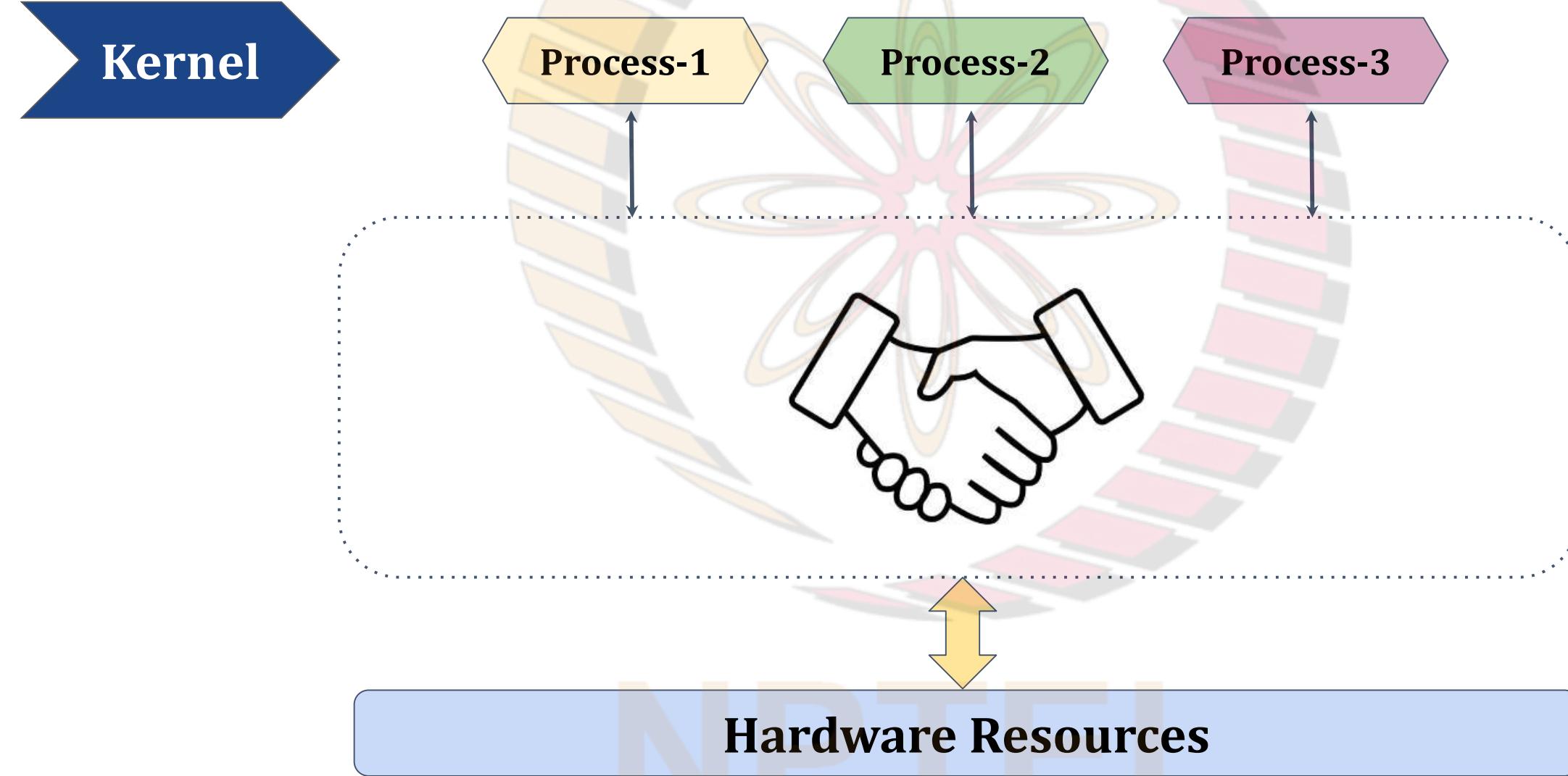
Process/Programs directly can't interact with H/W

Hardware Resources

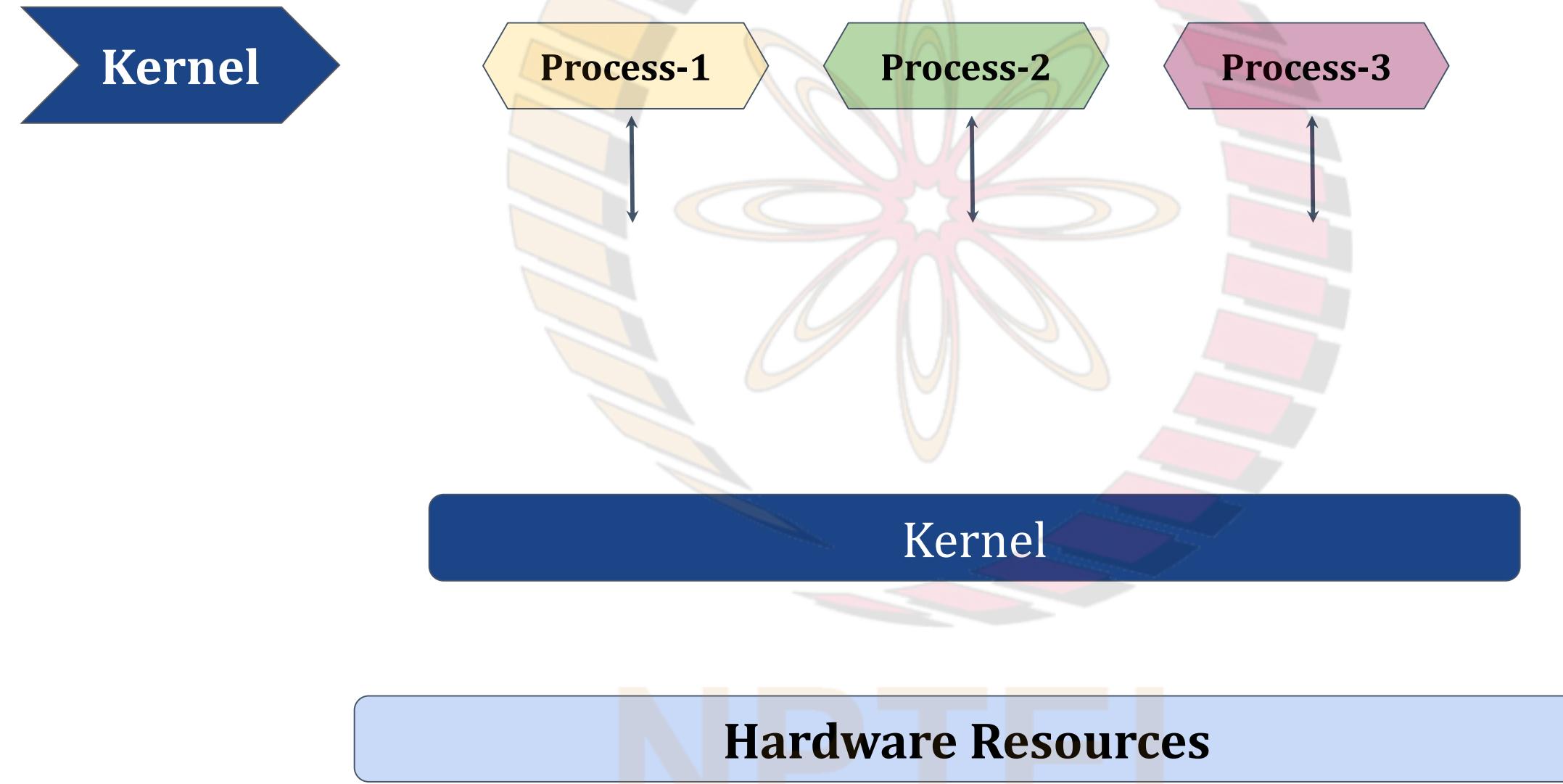
# Linux: Kernel



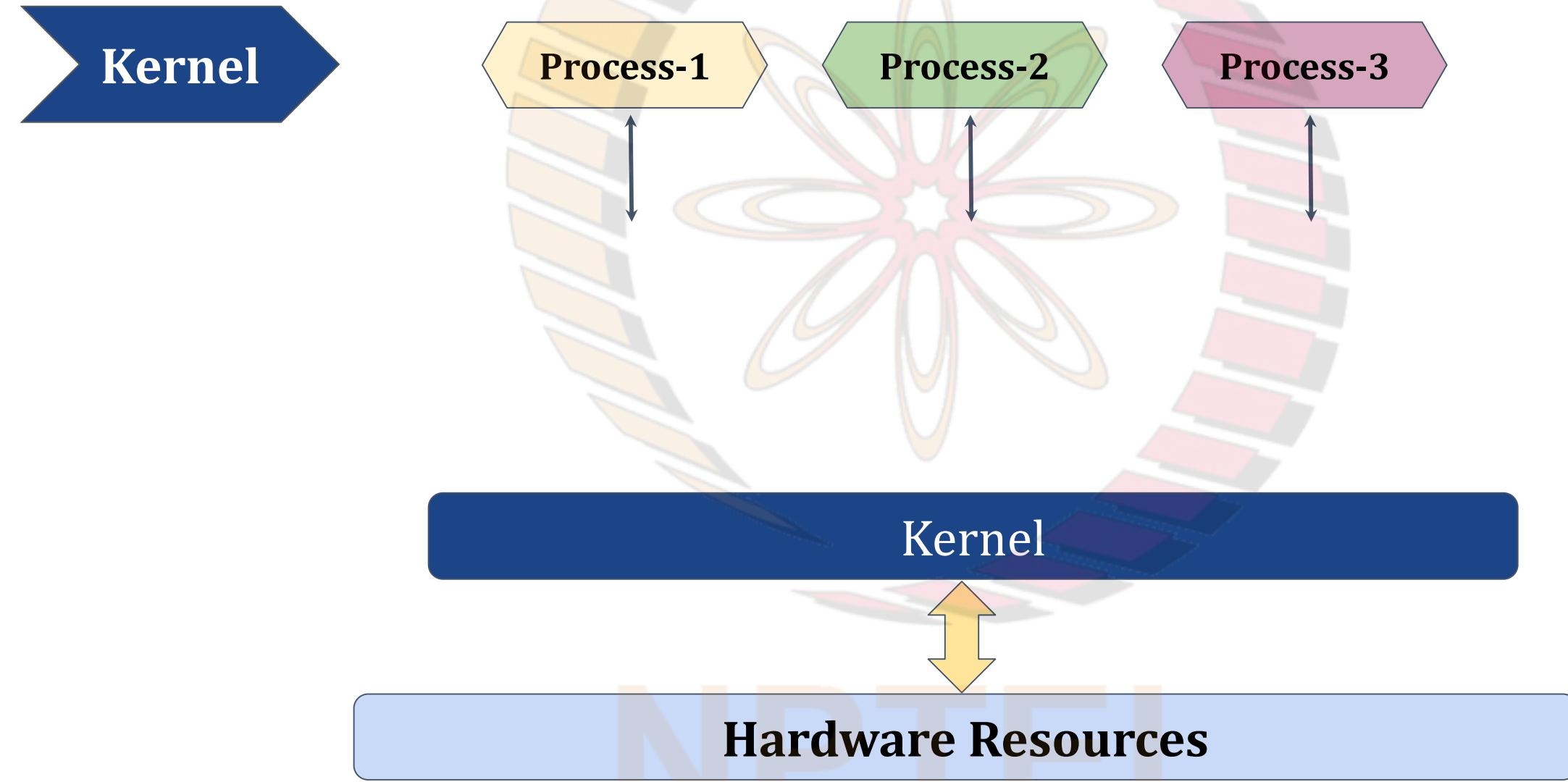
# Linux: Kernel



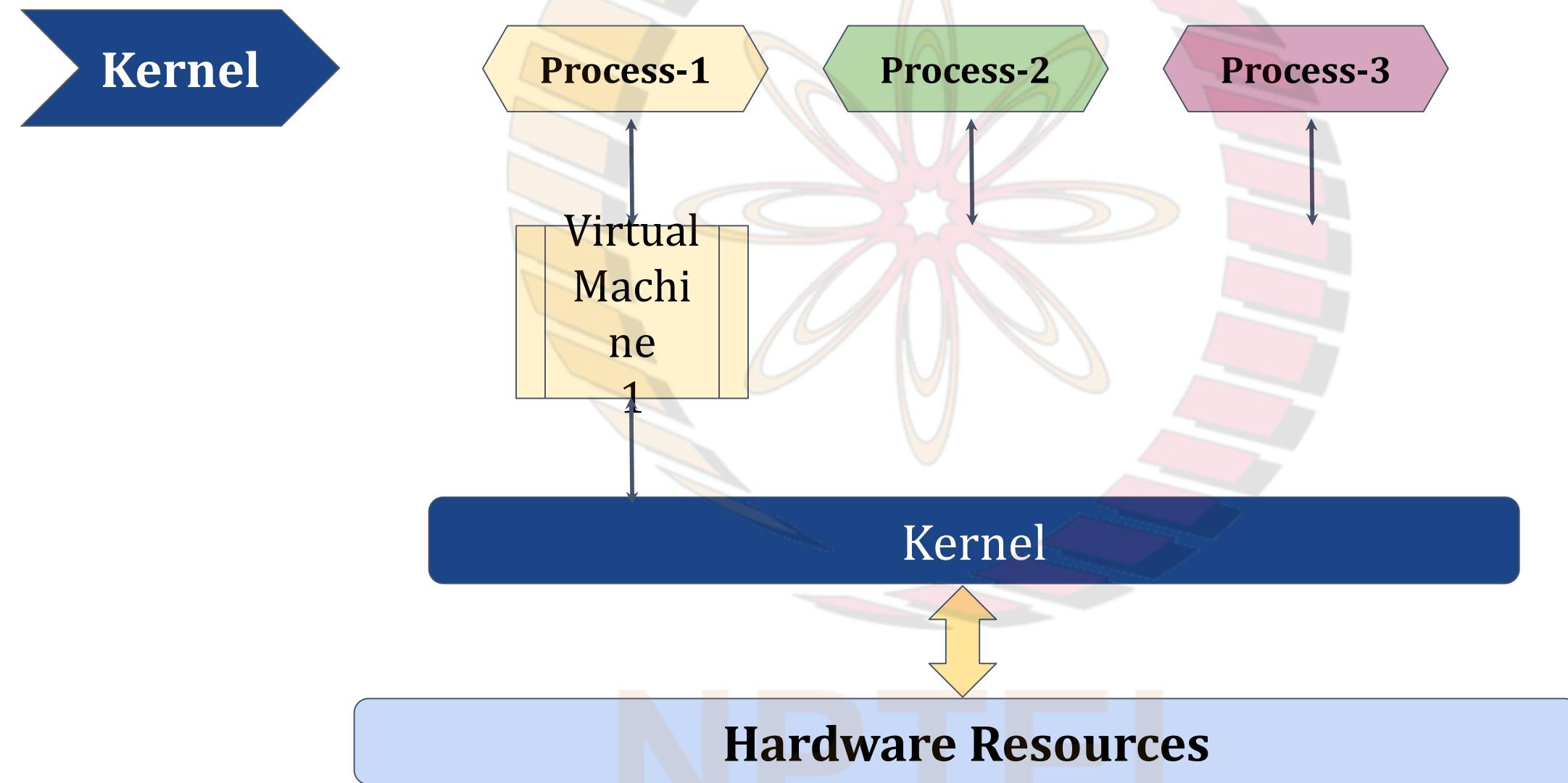
# Linux: Kernel



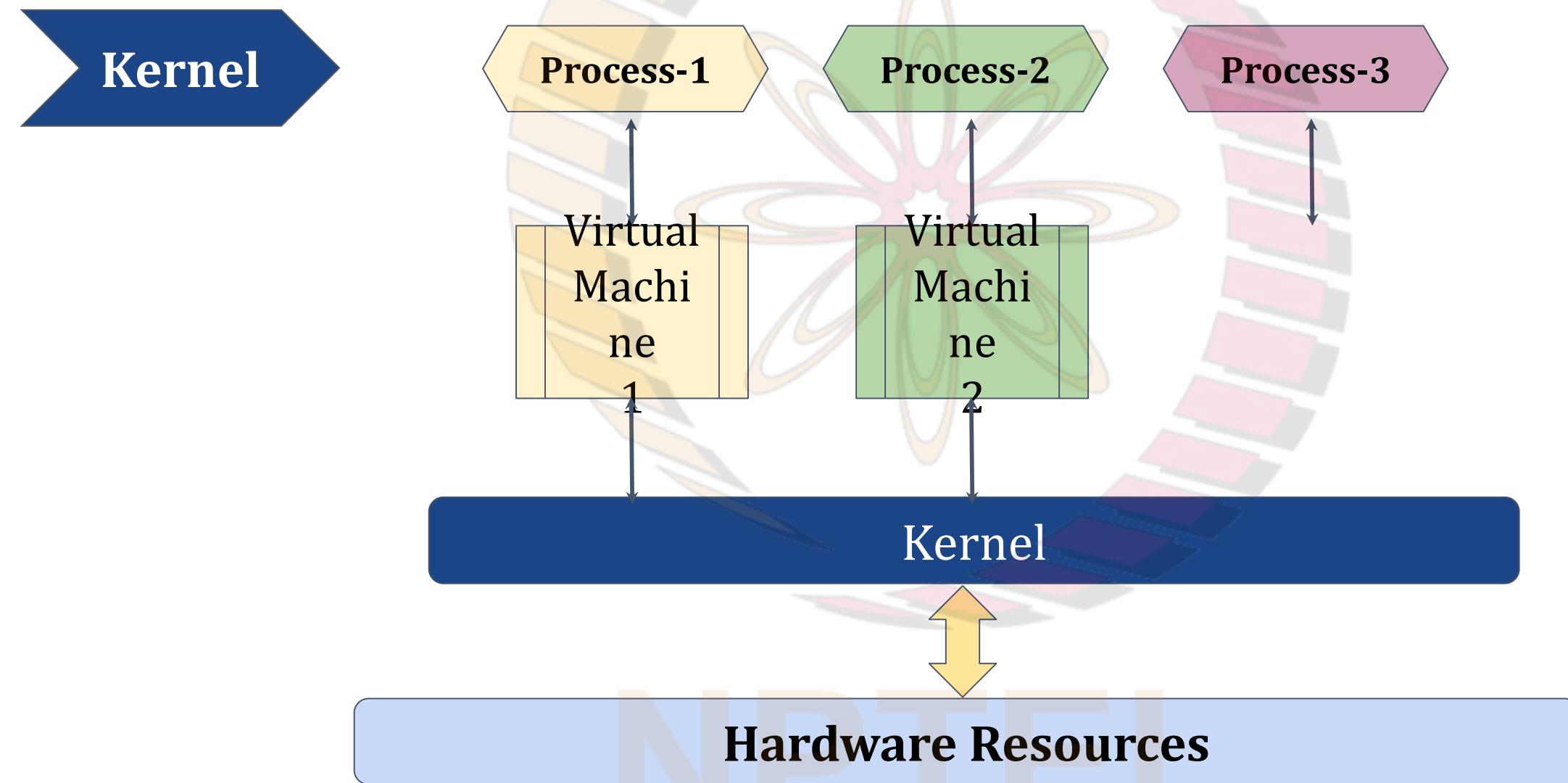
# Linux: Kernel



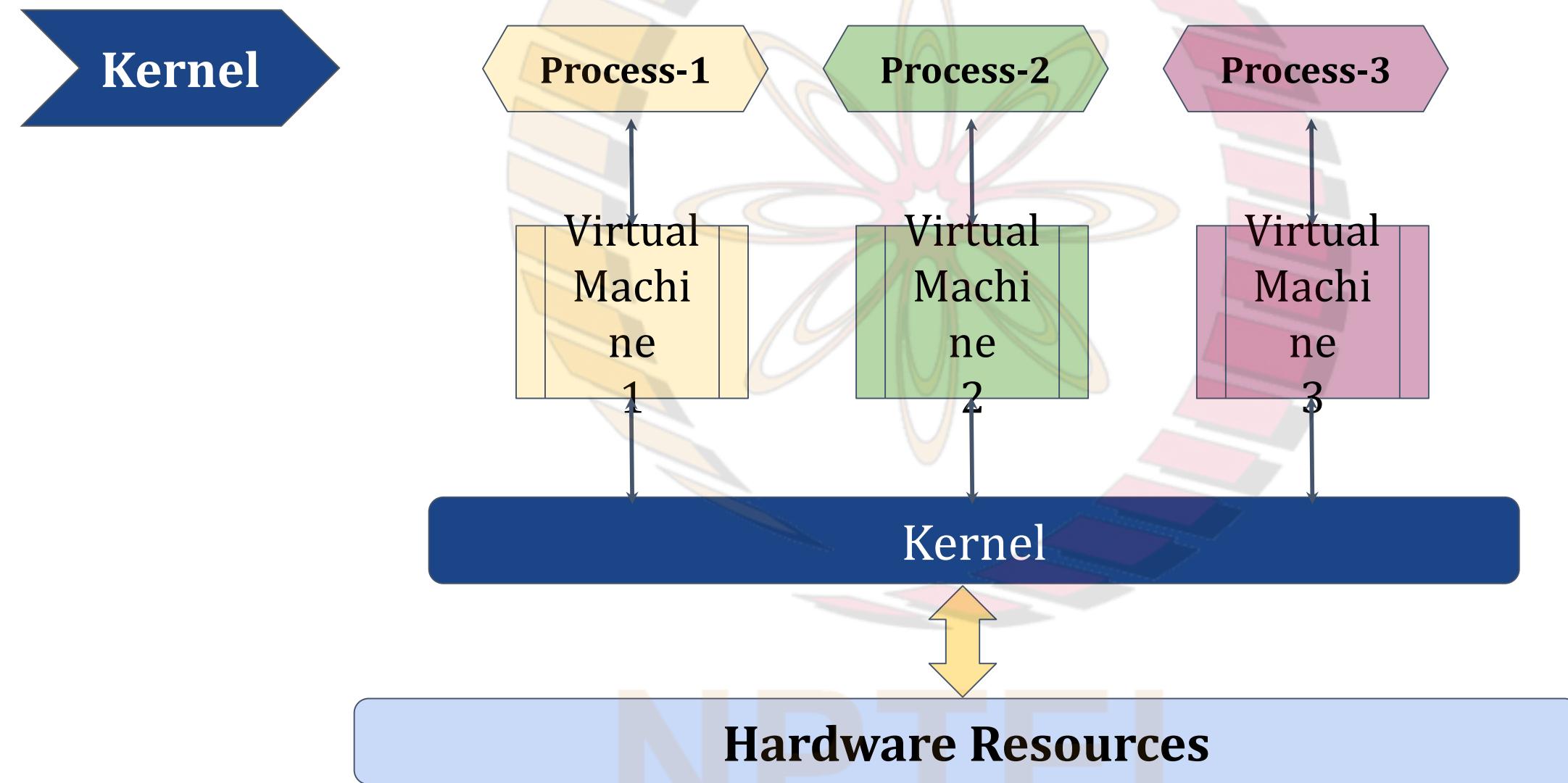
# Linux: Kernel



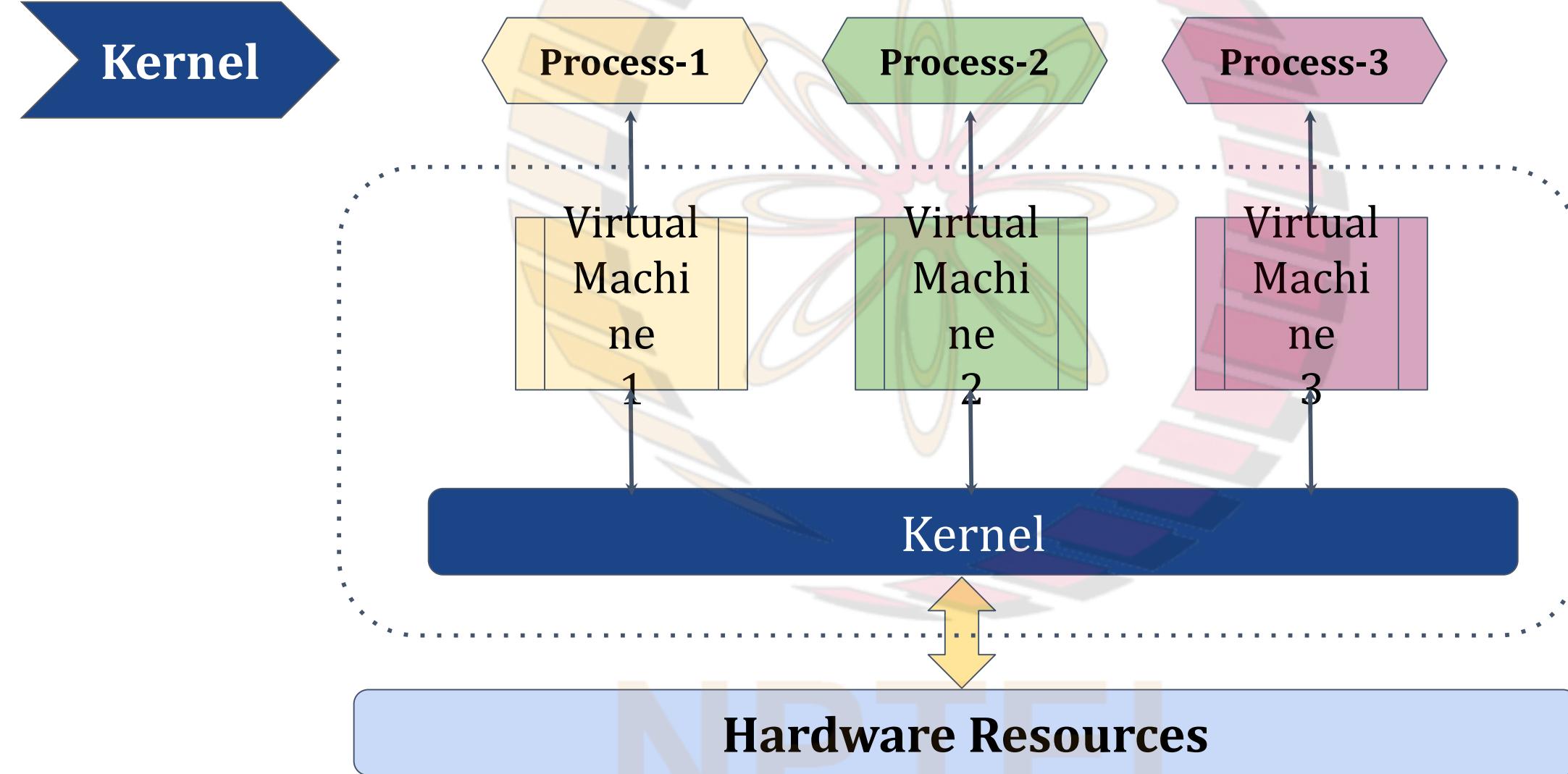
# Linux: Kernel



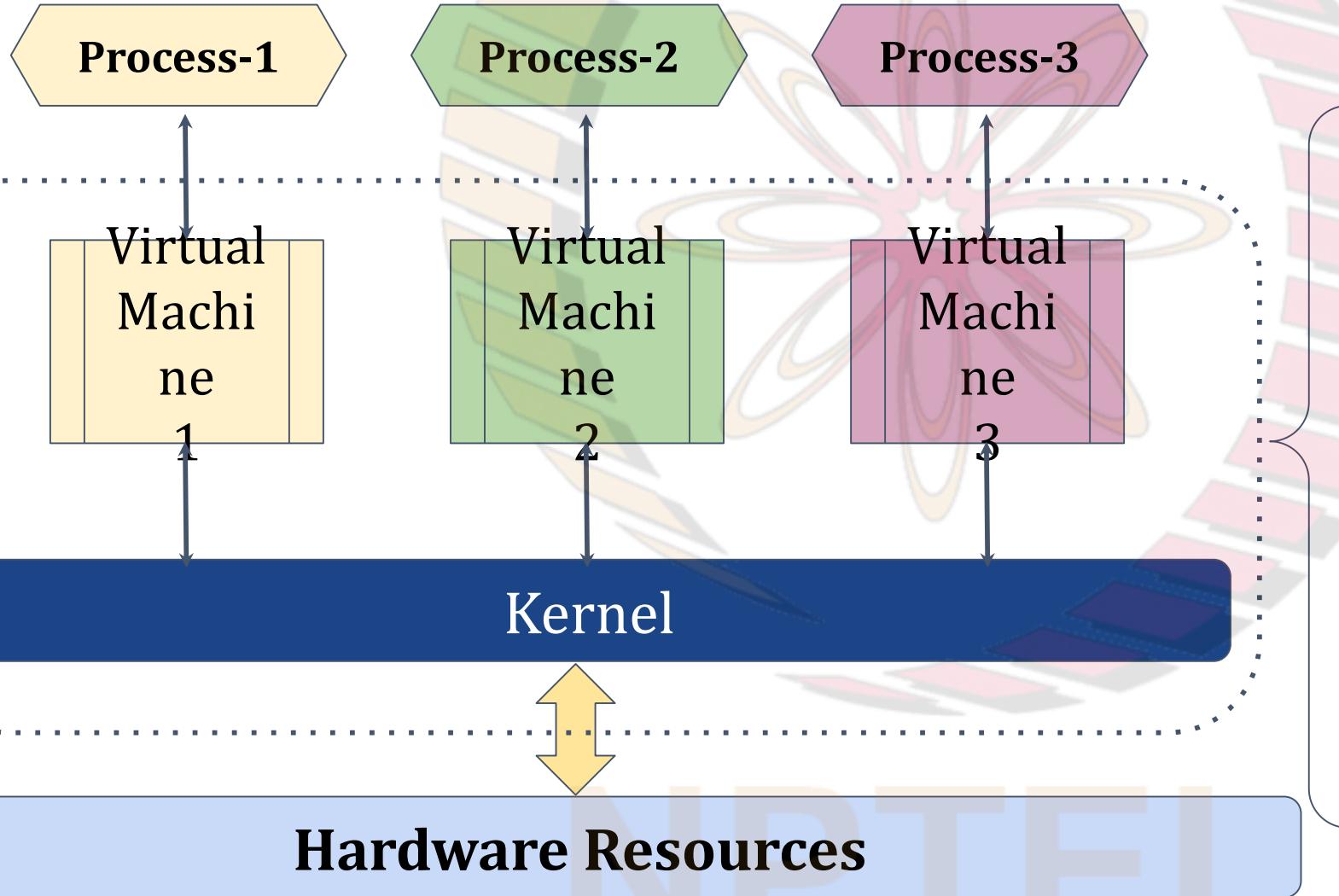
# Linux: Kernel



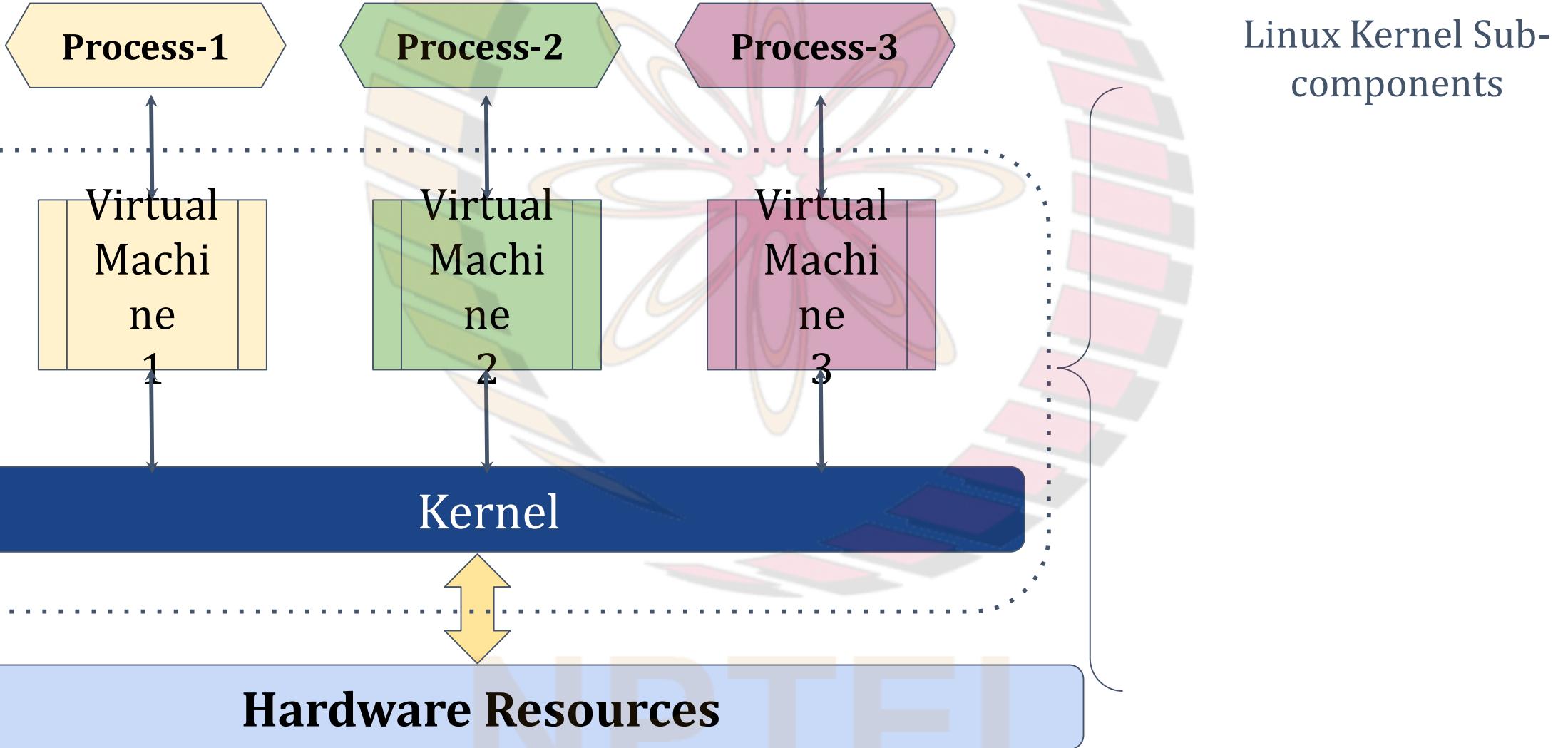
# Linux: Kernel



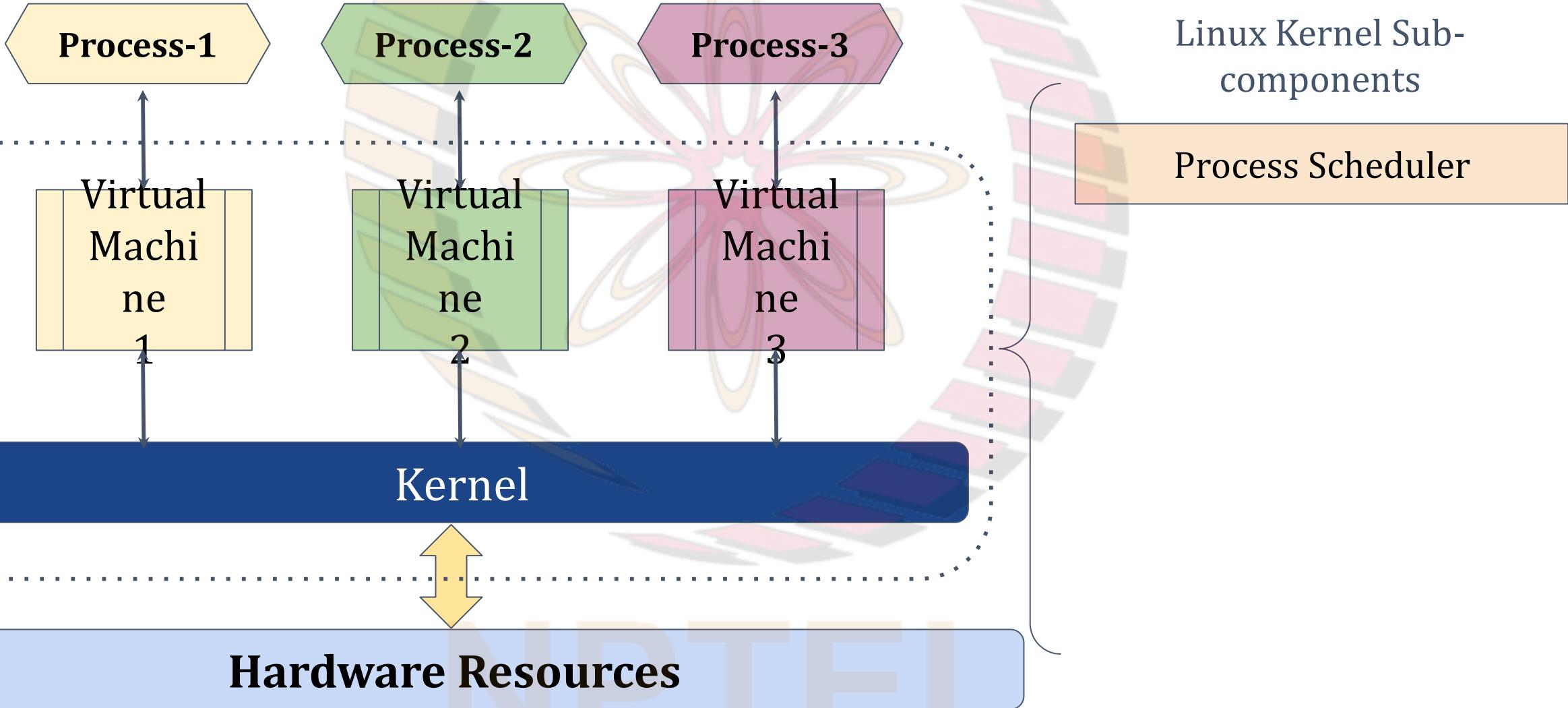
# Linux: Kernel



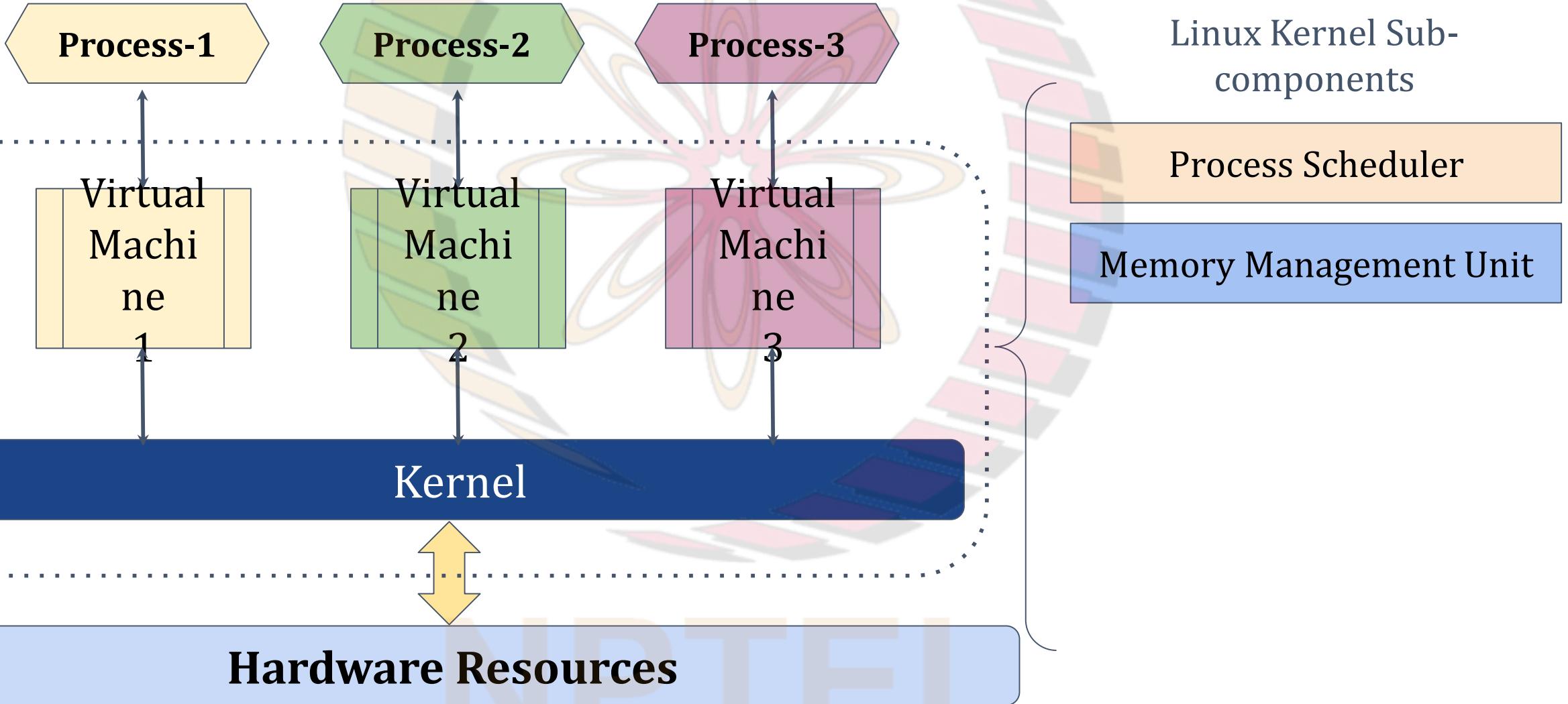
# Linux: Kernel



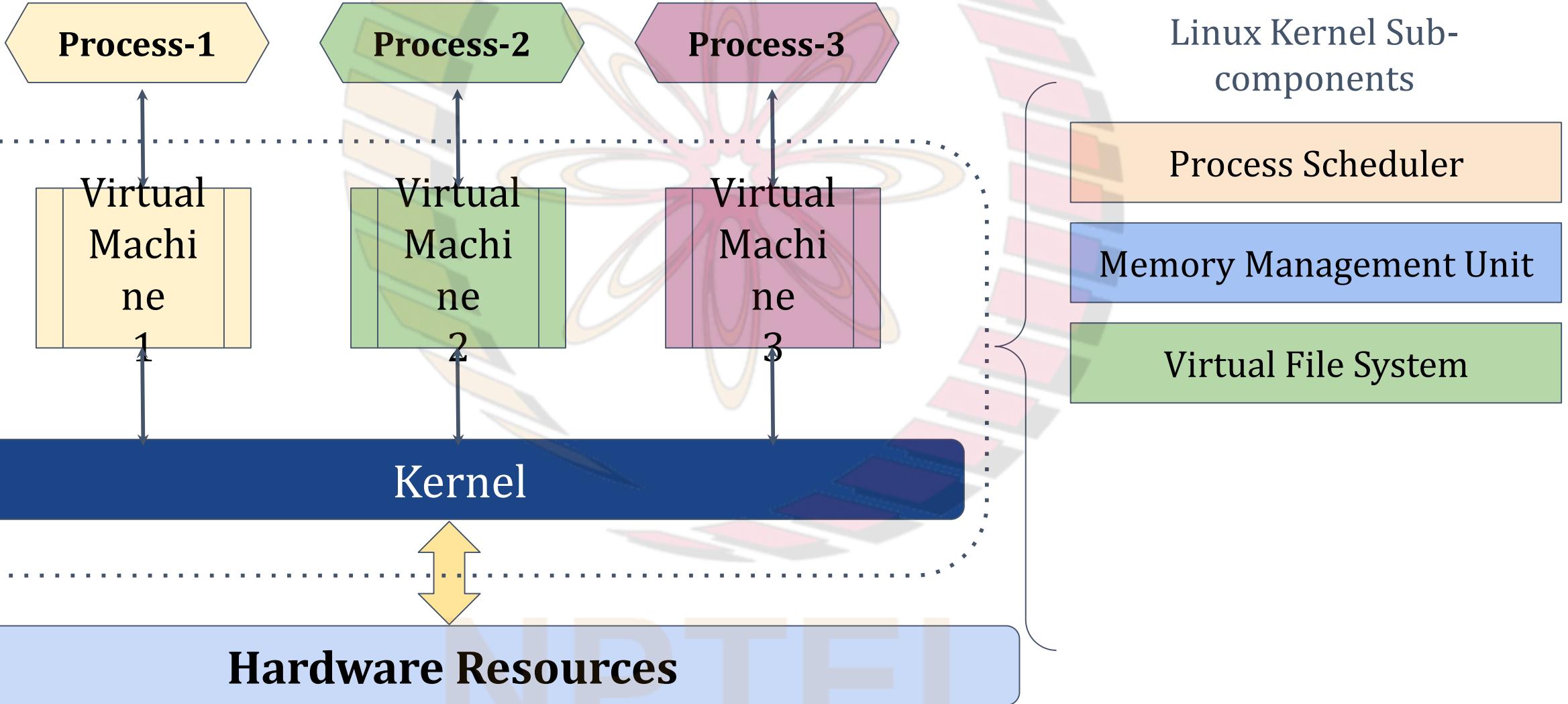
# Linux: Kernel



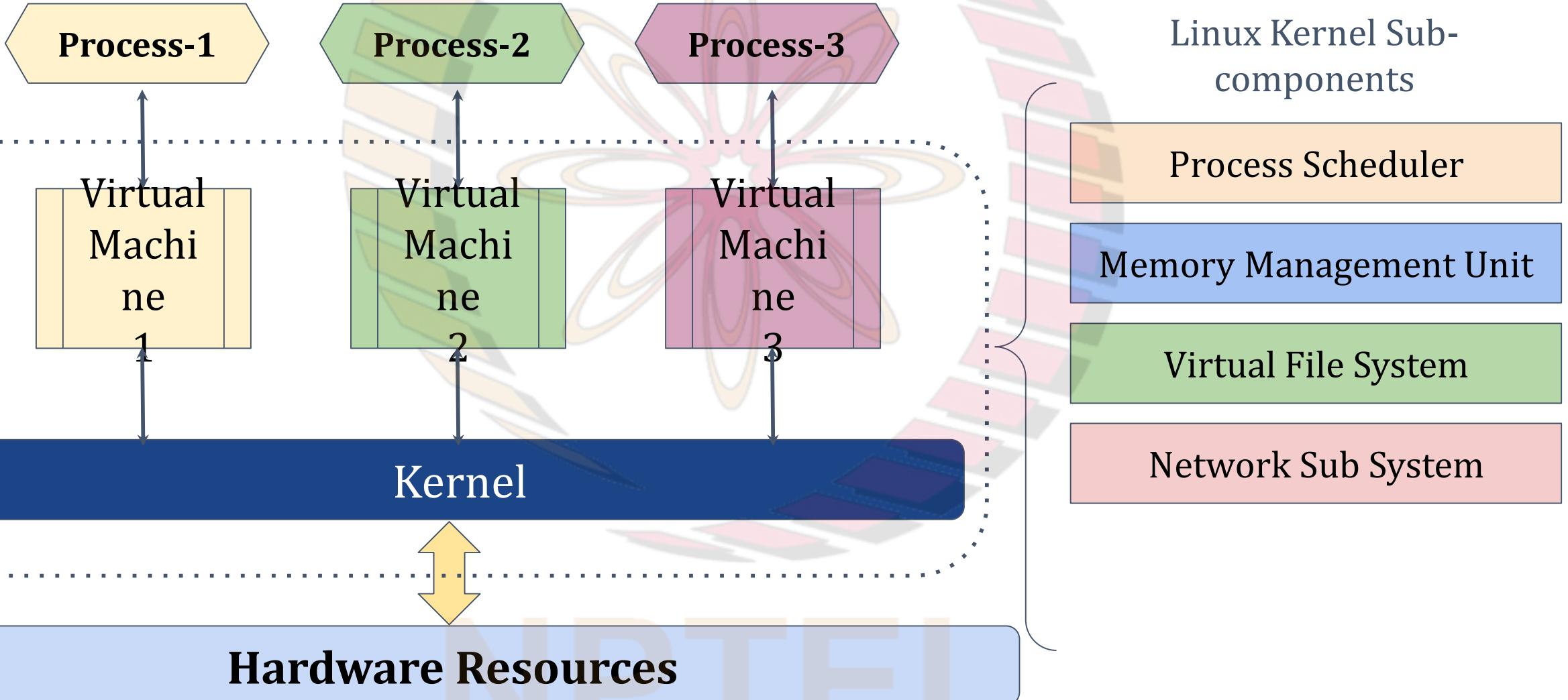
# Linux: Kernel



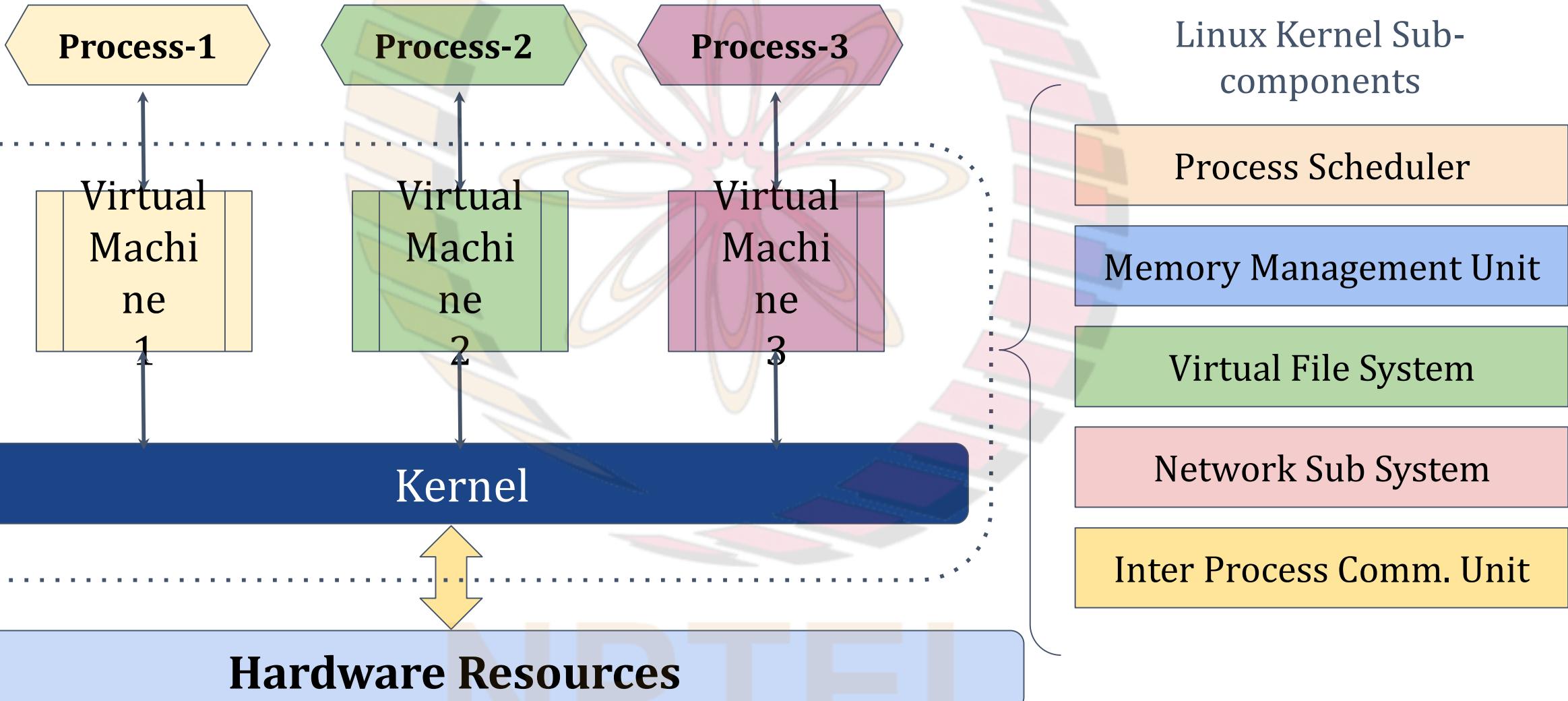
# Linux: Kernel



# Linux: Kernel



# Linux: Kernel





# Linux Kernel Sub- Components

NPTEL

# Linux: Kernel

1

Process Scheduler

NPTEL

# Linux: Kernel

1

## Process Scheduler

- It is responsible for managing how processes are allocated to the CPU
- Its primary goal is to maximize CPU utilization and ensure smooth multitasking by determining which processes get CPU time and for how long

# Linux: Kernel

1

## Process Scheduler

- It is responsible for managing how processes are allocated to the CPU
- Its primary goal is to maximize CPU utilization and ensure smooth multitasking by determining which processes get CPU time and for how long
- **Key Elements:**
  - Scheduling Policy: Round Robin(RR), First-Come-First-Served (FCFS), Shortest Job Next (SJN), Completely Fair Scheduler (CFS) etc
  - Process States: Running, Waiting, Ready, Blocket etc
  - Preemptive vs. Non-Preemptive Scheduling
  - Context Switching
  - Load Balancing

# Linux: Kernel

2

Memory Management Unit (MMU)

NPTEL

# Linux: Kernel

2

## Memory Management Unit (MMU)

- It is responsible for efficiently managing both **physical and virtual memory**, handling process memory allocations, and ensuring memory protection and isolation between processes

# Linux: Kernel

2

## Memory Management Unit (MMU)

- It is responsible for efficiently managing both **physical and virtual memory**, handling process memory allocations, and ensuring memory protection and isolation between processes
- **Key Elements:**
  - Address Translation (Virtual to Physical Mapping)
  - Page Tables - to store the mapping between virtual addresses and physical addresses
  - TLB (Translation Lookaside Buffer) - a small, fast cache to store recent data
  - Memory protection, Cache Control etc.

# Linux: Kernel

3

Virtual File System (VFS)

NPTEL

# Linux: Kernel

3

## Virtual File System (VFS)

- It is an abstraction layer that allows the Linux kernel to interface with different *file systems* in a uniform way.

# Linux: Kernel

3

## Virtual File System (VFS)

- It is an abstraction layer that allows the Linux kernel to interface with different ***file systems*** in a uniform way.
- **Key Elements:**
  - Superblock: Represents an entire mounted file system
  - Inode (Index Node): Stores metadata about individual files or directories
  - Dentry (Directory Entries): Represents directory entries
  - Mount Points: Where file systems are mounted
  - File System Operations, Types
  - File Descriptor Table: A table mapping file descriptors to open files for each process etc.

# Linux: Kernel

4

Network Sub-System

NPTEL

# Linux: Kernel

4

## Network Sub-System

- It manages communication between devices over a network.
- It is responsible for processing, routing, and managing all network packets as they pass through the system.

# Linux: Kernel

4

## Network Sub-System

- It manages communication between devices over a network.
- It is responsible for processing, routing, and managing all network packets as they pass through the system.
- **Key Elements:**
  - Socket API: interface between user-space applications and the kernel
  - Protocol: Link Layer(Ethernet, Wi-Fi), Network Layer(IP,IPv6,SSTP), Transport Layer (TCP, UDP), Application Layer (HTTP, FTP)
  - Network Stack, Network Devices -
  - Packet Processing, Routing, Traffic control etc.
  - ... so on

# Linux: Kernel

5

Inter Process Communication (IPC) Unit

NPTEL

# Linux: Kernel

5

## Inter Process Communication (IPC) Unit

- It allows processes to communicate and synchronize their actions with each other
- This communication is essential for tasks that require collaboration between multiple processes or need shared access to system resources.

# Linux: Kernel

5

## Inter Process Communication (IPC) Unit

- It allows processes to communicate and synchronize their actions with each other
- This communication is essential for tasks that require collaboration between multiple processes or need shared access to system resources.
- **Key Elements:**
  - Message Queues - allow processes to exchange messages
  - Shared memory - allows multiple processes to access the same memory region
  - Semaphores (Synchronization primitives), Signals (Asynchronous notifications) etc. ...



# Linux Library Files & Directory Structure

NPTEL

# Linux: Library Files



*"In the Linux Mostly Everything is a File"*

NPTEL

# Linux: Library Files



*"In the Linux Mostly Everything is a File"*

- In Linux, most system resources and devices are represented as ***files***
- This ***abstraction*** simplifies interaction with various components of the system

# Linux: Library Files



*"In the Linux Mostly Everything is a File"*

- In Linux, most system resources and devices are represented as **files**
- This **abstraction** simplifies interaction with various components of the system
- System libraries are **essential components** that provide commonly used functions and services to programs
- System libraries are **shared by** multiple applications, making them highly efficient in terms of memory usage and execution speed

# Linux: Types of Files



*"In the Linux Mostly Everything is a File"*

- **Regular files:** e.g. - text files, binary files, scripts

NPTEL

# Linux: Types of Files



*"In the Linux Mostly Everything is a File"*

- **Regular files:** e.g. - text files, binary files, scripts
- **Directories:** Directories themselves are treated as files that contain information about other files. e.g, /home/user/ is a file that contains information about files in the /home/user/ directory.

# Linux: Types of Files



*"In the Linux Mostly Everything is a File"*

- **Regular files:** e.g. - text files, binary files, scripts
- **Directories:** Directories themselves are treated as files that contain information about other files. e.g, /home/user/ is a file that contains information about files in the /home/user/ directory.
- **Devices:** Devices such as hard drives, keyboards, and printers are represented as files in the /dev directory.

# Linux: Types of Files



*"In the Linux Mostly Everything is a File"*

- **Symbolic Links:** A symbolic link (symlink) is a file that points to another file or directory

# Linux: Types of Files



*"In the Linux Mostly Everything is a File"*

- **Symbolic Links:** A symbolic link (symlink) is a file that points to another file or directory
- **Network Interfaces:** Network interfaces, like eth0 or wlan0, are treated like files within Linux

# Linux: Types of Files



*"In the Linux Mostly Everything is a File"*

- **Symbolic Links:** A symbolic link (symlink) is a file that points to another file or directory
- **Network Interfaces:** Network interfaces, like eth0 or wlan0, are treated like files within Linux

**EXCEPTION**



File concept doesn't strictly apply to ***processes, threads, or memory*** directly