

### Table of Contents



Diagram

- 1. Hardware Layer
  - 2. Kernel
  - 3. System Calls
  - 4. Shell
  - 5. User Space
  - 6. Libraries
- Benefits of Layered Architecture

RGPV PYQ

## Diagram



Linux operating system architecture

### 1. Hardware Layer

- Function: Provides the physical foundation upon which the Linux system operates.
- Components: CPU, memory, storage devices, peripherals.

## 2. Kernel

- Function: The core of the operating system, responsible for resource management, process control, and device communication.
- Components: Process scheduler, memory manager, device drivers, file system, network stack.

## 3. System Calls

- Function: Interface between user applications and the kernel.
- Example: `open()`, `read()`, `write()`, `close()`.

## 4. Shell

- Function: Command-line interpreter for interacting with the system.
- Types: Bourne Again Shell (`bash`), Z shell (`zsh`), C shell (`csh`).

## 5. User Space

- Function: Contains all user applications, libraries, and data.
- Components: Desktop environments, graphical user interfaces (GUIs), editors, games, utilities.

## 6. Libraries

- Function: Provide reusable code for common tasks, reducing development time.
- Types: C library (`glibc`), Standard Template Library (STL).

## Benefits of Layered Architecture

- Modular design: Facilitates independent development and maintenance of different layers.
- Flexibility: Allows for customization and adaptation to diverse hardware and software environments.
- Efficiency: Optimizes resource utilization by sharing functionality across layers.
- Portability: Enables Linux to run on a wide range of hardware platforms.

### References:

- “Operating Systems: Three Easy Pieces” by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (2018)
- “Linux Kernel Development” by Robert Love (2010)
- The Linux Kernel Archives: <https://archive.kernel.org/>