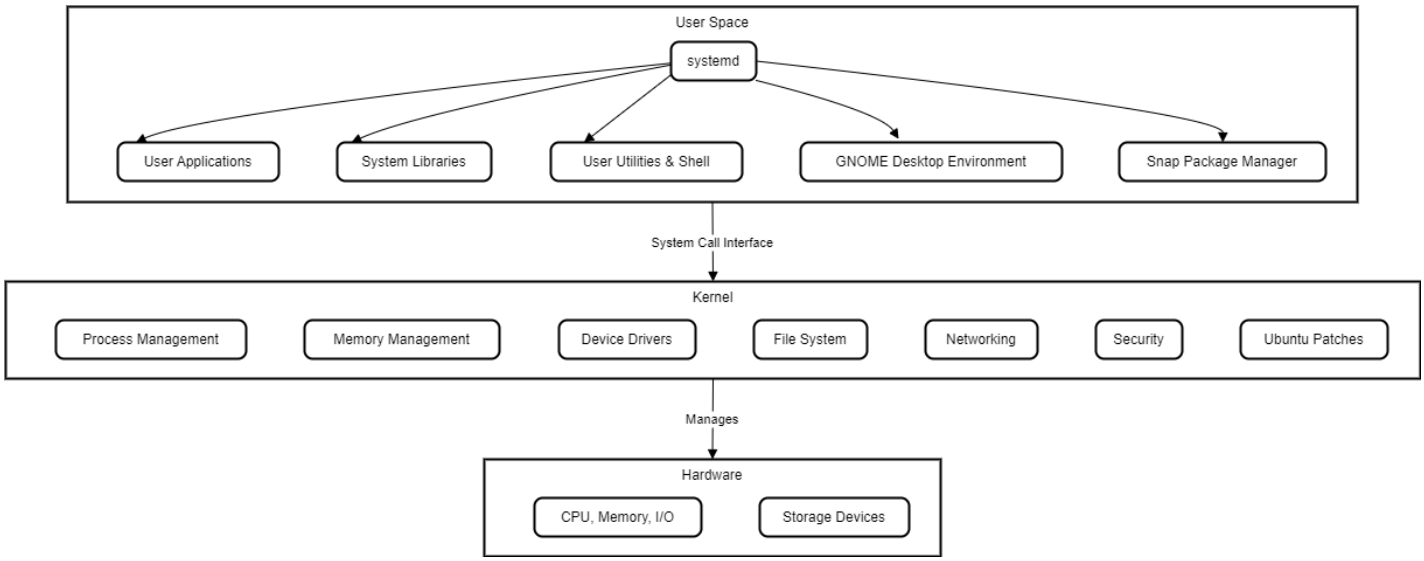
Exercise 3:

1. Linux internal architecture (ubuntu)



2. Explain the interactions between the kernel component & the user space component, group them in logical groups.

In a Linux operating system, the kernel and user space components work together to manage system resources and operations. The kernel acts as a bridge between the hardware and user space applications, providing essential services such as process management, memory management, device management, file system management, networking, and security. **User space** applications and utilities interact with the **kernel** through **system calls** to perform tasks, while the kernel ensures these requests are executed efficiently and securely. This collaboration ensures a seamless and efficient user experience on a Linux system.

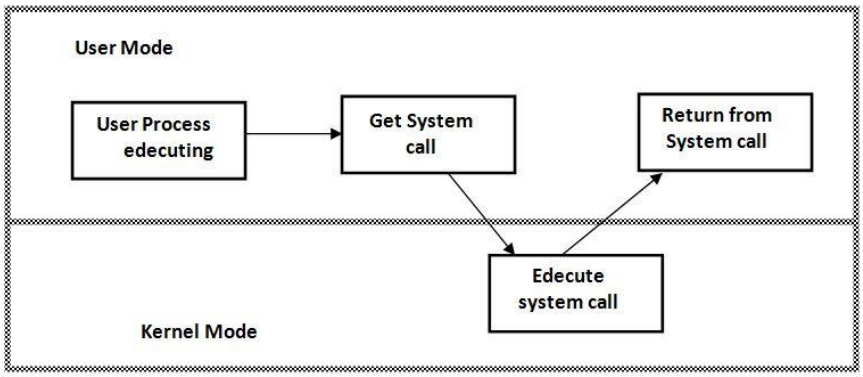


Figure : System calls

Examples:

1. Process Management

The kernel manages processes.

The user space send system calls to kernel:

- User Applications**:** When a user runs a program, the kernel creates a process for it using system calls like `fork()`, `exec()`, `wait()`, and `kill()`.
  Example: A web browser started so a process is created and managed by the kernel.
- System Utilities**:** Commands such as `ps`, `top`, and `kill` interact with the kernel to display and manage running processes.
  Example: Using `ps` to list all active processes.

2. Memory Management

The kernel manages memory.

The user space send system calls to kernel:

- User Applications**:** Applications request and manage memory through system calls like `malloc()`, `free()`, `mmap()`, and `munmap()`.
  Example: A text editor allocates memory to open and edit a document.
- System Libraries**:** Libraries like `glibc` use these system calls to handle memory allocation.
  Example: `glibc` allocates memory for a new string in a C program.