8. Research the Linux kernel's handling of Ethernet devices and network interfaces. Write a short report on how the Linux kernel supports Ethernet communication (referencing kernel.org documentation).

The Linux network stack follows a layered architecture, where Ethernet operates at Layer 2 (Data Link Layer). The net _ device structure represents a network interface, managed by the kernel.

The Linux kernel provides robust support for Ethernet communication through a structured network stack. The stack follows a layered architecture, with Ethernet operating at Layer 2 (Data Link Layer). The kernel abstracts network interfaces using the net_device structure, which plays a central role in managing network interfaces, including Ethernet devices.

Network Stack Architecture

The Linux network stack is divided into multiple layers, adhering to the OSI model. Ethernet communication primarily occurs at the Data Link Layer but interacts with higher layers such as the Network Layer (IP) and Transport Layer (TCP/UDP). The key components of the network stack involved in Ethernet communication include:

1. **Hardware and Drivers**: Ethernet drivers interact with network interface cards (NICs) and register themselves with the kernel.
2. **net_device Structure**: Represents a network interface, containing function pointers for transmission, reception, and configuration.
3. **Packet Processing Pipeline**:
    - Incoming packets traverse the kernel via network device drivers.
    - The netif_rx() function hands the packet to higher layers for processing.
    - Outgoing packets are transmitted using the ndo_start_xmit() function.
4. **Socket Interface**: Applications interact with network interfaces through sockets, which provide an abstraction over low-level Ethernet communication.

net_device Structure

The net_device structure is a core abstraction in the Linux kernel that represents a network interface. It includes fields and function pointers that define how packets are transmitted and received. Key fields include:

- name: Interface name (e.g., eth0).
- mtu: Maximum Transmission Unit.
- flags: Status and configuration flags.
- netdev_ops: Pointer to operations such as ndo_open, ndo_stop, and ndo_start_xmit.

Network device drivers register their devices using register_netdev(), linking the hardware with the kernel's network stack.

Ethernet Frame Processing

When an Ethernet frame arrives:

1. The NIC receives the frame and places it into the kernel’s buffer.
2. The driver notifies the kernel via an interrupt.
3. The kernel calls netif_rx() to pass the frame to the networking stack.
4. The frame is processed by protocols like ARP, IP, or custom handlers.

For outgoing packets:

1. The kernel routes the packet to the appropriate network interface.
2. The ndo_start_xmit() function of the net_device structure sends the frame to the NIC.
3. The NIC transmits the frame onto the network.