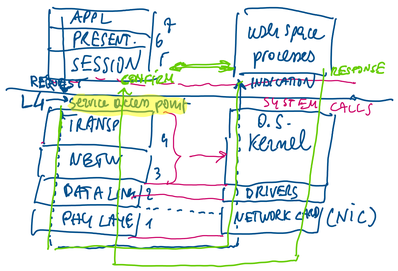
3.1 OSI SAP AND MAPPING ON O.S.

martedì 5 marzo 2024 08:33



- blocking functions implement the asynchronous events by returning asynchronously
- blocking functions are easy to manage but cannot support multiple handling
- Multi-thread solve this problem but introduces overheads plus possible criticalities in the program

- interrupt/signal handlers are the natural and straightforward implementation of asynchronous signal handling (e.g. indication, confirm).

Understanding the interaction types through Service Access Points (SAPs) at the OSI model's Layer 7 (the Application Layer) can be illustrated using an HTTP request example. The Application Layer provides network services directly to end-user applications. The four main types of interactions are Request, Indication, Response, and Confirm. Here's how these interactions can occur in the context of an HTTP request:

1. **Request**
- Description: A "Request" operation at Layer 7 is initiated by an application when it needs to use a network service. For an HTTP request, the web browser (acting as a client) generates an HTTP GET request to retrieve a webpage.
- Practical Example: The browser sends an HTTP GET request to the server to ask for the HTML content of a webpage.

2. **Indication**
- Description: An "Indication" occurs when the server receives the HTTP GET request. This informs the server application (such as a web server) that there is an incoming request that needs to be processed.
- Practical Example: The web server receives the HTTP GET request and is now aware that a client is requesting the content of a specific webpage.

3. **Response**
- Description: A "Response" is generated by the server application in reply to the received indication. After processing the HTTP GET request, the server prepares an HTTP Response containing the requested webpage.
- Practical Example: The web server responds to the browser's request by sending an HTTP Response that includes the HTML content of the requested webpage.

4. **Confirm**
- Description: In the context of HTTP and Layer 7 interactions, a "Confirm" operation is implicitly handled within the response itself, indicating to the client application (the browser) that its request has been processed and a response is being provided.
- Practical Example: Once the browser receives the HTTP Response, the retrieval of the webpage can be considered confirmed. The browser then renders the webpage for the user.

This sequence of operations illustrates how Layer 7 applications use the OSI model's service primitives to communicate over a network. In the case of an HTTP request, the process involves a web browser requesting a webpage and a web server processing that request and responding with the desired content. The seamless flow from request to confirmation showcases the effectiveness of these interactions in enabling complex web communications.

Mapping the OSI model layers onto the components of an operating system (OS) illustrates how network communication is facilitated through different levels of abstraction, from user applications down to the physical transmission of data. This mapping involves user processes, kernel services, device drivers, and the hardware Network Interface Card (NIC). Here's a generalized view of how the OSI layers correlate with these OS components:

User Processes (Application, Presentation, and Session Layers)
- OSI Layers: Application (Layer 7), Presentation (Layer 6), and Session (Layer 5).
- Description: These layers are typically implemented in software as part of user processes. They handle high-level networking functions, including data formatting, encryption/decryption, authentication, and session management.
- Mapping: Applications running in user space directly interact with the network through system calls that communicate with the underlying OS networking stack. Examples include web browsers, email clients, and file transfer applications.

Kernel Services (Transport and Network Layers)
- OSI Layers: Transport (Layer 4) and Network (Layer 3).
- Description: The Transport layer provides end-to-end communication services for applications, such as TCP/UDP port management, flow control, and error checking. The Network layer is responsible for routing and forwarding packets across different networks.
- Mapping: These functions are typically implemented within the OS kernel. The kernel manages networking protocols (e.g., TCP, UDP, IP) and performs tasks such as packet routing, fragmentation, and reassembly.

Device Drivers (Data Link Layer)
- OSI Layer: Data Link (Layer 2).
- Description: This layer provides node-to-node data transfer—a link between two directly connected nodes. It handles framing, physical addressing (MAC addressing), and error detection/correction.
- Mapping: Device drivers operate as an interface between the OS kernel and the hardware. They translate generic networking commands from the kernel into hardware-specific operations. These drivers manage how data packets are encoded and transmitted over the network medium.

Hardware NIC (Physical Layer)
- OSI Layer: Physical (Layer 1).
- Description: The Physical layer is concerned with the physical transmission of raw bit streams over the communication medium. It defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link.
- Mapping: The Network Interface Card (NIC) and its associated hardware (such as Ethernet cables and connectors) embody the Physical layer. The NIC, controlled by its device driver, handles the actual transmission and reception of data on the network medium.

Summary of the Mapping
- User Processes: Interact with the network through high-level protocols implemented in applications (OSI Layers 5-7).
- Kernel Services: Core networking functionalities, including routing and transport protocols, are managed within the OS kernel (OSI Layers 3-4).
- Device Drivers: Act as intermediaries between the kernel's networking stack and the physical hardware, managing data packet preparation for transmission (OSI Layer 2).
- Hardware NIC: The physical interface for data transmission, implementing the specifications of the Physical layer (OSI Layer 1).