**Reactor Pattern**

* **Step 1:** Server starts the Reactor and listens for connections.
* **Step 2:** Client sends a connection request.
* **Step 3:** Reactor detects the new connection and accepts it.
* **Step 4:** Client sends data (request).
* **Step 5:** Reactor detects client activity and invokes the handler.
* **Step 6:** Server processes the request and (optionally) sends a response.
* **Step 7:** Reactor continues monitoring all clients and handles new requests.

This loop continues, allowing the server to handle thousands of clients simultaneously without blocking, efficiently processing requests as they arrive.

**Steps for Working with Proactor and Reactor Together:**

* **Step 1:** The Reactor monitors the sockets using **select()** to detect when they are ready.
* **Step 2:** When a socket is ready (e.g., a client sends a request), the Reactor detects it.
* **Step 3:** Instead of handling the operation synchronously, the Reactor delegates the handling to the Proactor.
* **Step 4:** The Proactor creates a **thread** to handle the operation in the background (such as reading or writing).
* **Step 5:** After the operation is completed, the thread terminates, and the Reactor continues monitoring other connections.

ראש הטופס

תחתית הטופס