

Application Protocol

1. Client-Server Interaction:

- **Client Sends:** Requests for displaying the catalog, searching for a book by title or ISBN, ordering a book, paying for an order, and terminating the connection.
- **Server Responds:** Results of each operation (success or failure messages, book details, order status, etc.).

2. Message Formats:

- **Request Messages (Client to Server):**
 - **1:** Display catalog request.
 - **2 title:** Search book by title request followed by the title.
 - **2 ISBN:** Search book by ISBN request followed by the ISBN.
 - **3 ISBN:** Order book request followed by the ISBN.
 - **4 order_number:** Pay for book request followed by the order number.
 - **5:** Terminate connection request.
 - **Response Messages (Server to Client):**
 - For **1, 2, 3, 4:**
 - **"found"** or **"not found"** status indicating success or failure.
 - Book details (if found).
 - Order number (for new orders).
 - For **5:**
 - Connection termination acknowledgment.
 - **Error Handling:**
 - Messages indicating errors in case of failed operations (e.g., file not found, book not available, order not found).
- ## 3. Data Formats:
- **Book Structure:**
 - Serial number, title, authors, ISBN, publisher, date of publication.
 - **Order Structure:**
 - Order number, ISBN, status (**unpaid** or **paid**).

Conceptual Server Algorithm

1. Initialization:

- Create a socket and bind it to a specific port.
- Start listening for incoming connections.
- Accept client connections and spawn a new thread for each client to handle its requests concurrently.

2. Thread Handling (**handle_client** function):

- Upon client connection, read the client's request (choice of action).
- Switch based on the request type (1 to 5):
 - **Case 1 (Display catalog):**
 - Read book details from "bookfile.txt".
 - Send the count of books and book details to the client.
 - **Case 2 (Search book):**
 - Based on client choice (title or ISBN), read search criteria.
 - Search for the book in memory.
 - Send search result (found or not found) and book details (if found) to the client.
 - **Case 3 (Order book):**
 - Read the ISBN of the book to order.
 - Generate an order number and store order details in "orderfile.txt".
 - Send order status (found) and order number to the client.
 - **Case 4 (Pay for book):**
 - Read the order number to process payment.
 - Update payment status (paid) in "orderfile.txt".
 - Send payment status (found or not found) to the client.
 - **Case 5 (Terminate connection):**
 - Close the client socket and terminate the thread.

3. File Operations:

- Use file operations (fopen, fclose, fprintf, fscanf) to read from and write to "bookfile.txt" and "orderfile.txt" for storing book and order information.

4. Error Handling:

- Handle errors gracefully:
 - Display appropriate error messages (perror) and continue processing or terminate the connection/thread based on the severity of the error.

5. Cleanup:

- Close file descriptors and sockets appropriately to ensure clean shutdown.