
Subject Name: Computer Networks

Subject Code: CS6111

Lab Exercise: 03

Spot

Scenario:

You have a server and two clients in a networked environment. The server is designed to handle multiple numbers sent by clients, store them in a buffer, and process them under specific conditions. This scenario involves handling buffer overflow, sorting numbers, and responding to clients about the nature of their numbers.

Detailed Implementation Steps [Look at Example Communication Flowfirst]

1. Server Implementation

1. Initialize the Server

- Create a buffer with a fixed size (e.g., 10 numbers).

2. Handle Incoming Requests

- **Receive Number:** Accept numbers from clients.
- **Buffer Management:**
 - **Add Number:** Store the number in the buffer.
 - **Check Capacity:** If the buffer exceeds its limit, notify the client that the buffer is full (e.g., send "Buffer Exceeded" message).

3. Buffer Overflow Handling

- **Sort Numbers:** Once the buffer is full, sort the numbers in ascending order.
- **Determine Smallest Number:** Extract the smallest number from the sorted buffer.
- **Check Armstrong Number:**
 - Implement a function to check if a number is an Armstrong number (e.g., 153 is an Armstrong number).

4. Respond to Clients

- **Smallest Number:** Respond to the client with the smallest number with a message indicating if the number is an Armstrong number or not.
- **Remove Processed Number:** After responding, remove the smallest number from the buffer.
- **Continue Processing:** Repeat the process for the next smallest number until all numbers have been processed.

5. Error Handling

-
- **Buffer Exceeded:** Handle situations where the buffer exceeds its capacity by notifying clients and managing the buffer state.

2. Client Implementation

1. Send Numbers

- **Client 1 & Client 2:** Send numbers to the server at different times.
- **Multiple Sends:** Clients can send multiple numbers in sequence or at different intervals.

2. Receive Response

- **Handle Server Response:** Process responses from the server, which include whether a number is an Armstrong number or not.
- **Display Results:** Display the result to the user or handle it according to the application's requirements.

Example Communication Flow

1. **Client 1** sends number 9474 to the server.
 2. **Client 2** sends number 153 to the server.
 3. **Server** stores these numbers in the common buffer every time a client sends a number.
 4. **Client 1** then sends number 370 to the server.
 5. **Server** buffer now contains [153, 9474, 370]. [Assume the buffer size is 3]
 6. **Server** checks if the buffer is full. If it is, it sorts the numbers: [153, 370, 9474].
 7. **Server** responds to **Client 2** (which sent the smallest number, 153) whether the number is an Armstrong number.
 8. **Server** removes 153 from the buffer and continues with the next smallest number, 370.
 9. **Server** responds to **Client 1** (which sent 370) and so on.
 10. **Server** removes 370 from the buffer and continues with the next smallest number, 9474.
 11. **Server** responds to **Client 1** (which sent 9474) and once the buffer is empty, the server closes all the connections and terminates.
-