|  |  |
| --- | --- |
|  | **DEPARTMENT OF COMPUTER ENGINEERING** |

Experiment No. 02

|  |  |
| --- | --- |
| Semester | B.E. Semester VIII – Computer Engineering |
| Subject | Distributed Computing Lab |
| Subject Professor In-charge | Dr. Umesh Kulkarni |
| Assisting Professor | Prof. Prakash Parmar |
| Academic Year | 2024-25 |

|  |  |
| --- | --- |
| Student Name | Deep Salunkhe |
| Roll Number | 21102A0014 |

**Title:** Concepts of Operating Systems in Distributed Computing

**Explanation:**

The objective of this lab was to implement a distributed application using socket programming in Java. The application consists of a **server** that calculates the factorial of an integer received from the **client** and sends the result back. This exercise helped demonstrate the fundamental principles of client-server communication, socket programming, and distributed system design.

**Concepts Covered:**

1. **Distributed Computing**: Distributed computing involves distributing tasks across multiple computers that communicate with each other over a network. In this lab, a client sends a request to the server for a factorial calculation, and the server computes the result and returns it. This interaction between the client and server is a fundamental example of distributed computing, where each component (client and server) operates on separate systems (in this case, they could even be on different machines, though here they are assumed to be local).
2. **Client-Server Architecture**: A **client-server architecture** is a common design pattern in distributed systems. In this model, the **server** provides a service or resource, while the **client** accesses this service. In the application implemented in this lab:
   * The **server** listens on a specific port for incoming client requests, processes them, and sends back a response.
   * The **client** sends an integer to the server, requests the factorial calculation, and receives the computed result.
3. **Socket Programming**: Socket programming allows communication between two computers over a network by using a combination of IP addresses and ports. In this lab, we used **Java Sockets** to facilitate the client-server communication:
   * The **ServerSocket** class in Java is used to create a server-side socket that listens for incoming client connections.
   * The **Socket** class is used on the client side to establish a connection with the server.

Sockets provide a way for applications to exchange data over a network, making them a core tool for implementing distributed applications.

1. **Factorial Calculation**: The server in this lab receives an integer from the client, calculates its **factorial**, and sends the result back. A factorial of a number nnn is the product of all positive integers less than or equal to nnn. It is defined as:

n!=n×(n−1)×(n−2)×⋯×1n! = n \times (n-1) \times (n-2) \times \dots \times 1n!=n×(n−1)×(n−2)×⋯×1

For example:

* + 5!=5×4×3×2×1=1205! = 5 \times 4 \times 3 \times 2 \times 1 = 1205!=5×4×3×2×1=120
  + 3!=3×2×1=63! = 3 \times 2 \times 1 = 63!=3×2×1=6

In the code, a method (calculateFactorial) implements this logic using a simple loop to multiply numbers from 1 to the given integer.

**CODE:  
  
A computer screen with white text

Description automatically generated**

**A blue screen with white text

Description automatically generated**

**Output:**

