**1. TokenRingServer.java**

The server will hold and pass the token between clients in the ring.

java

CopyEdit

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class TokenRingServer {

private static final int PORT = 12345;

private static List<Socket> clientSockets = new ArrayList<>();

private static int clientCount = 2; // Number of sites

public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket(PORT);

System.out.println("Token Ring Server started...");

// Accept clients

for (int i = 0; i < clientCount; i++) {

Socket clientSocket = serverSocket.accept();

clientSockets.add(clientSocket);

System.out.println("Client " + (i + 1) + " connected.");

}

// Create a thread to pass the token

Thread tokenPassingThread = new Thread(() -> passToken());

tokenPassingThread.start();

} catch (IOException e) {

e.printStackTrace();

}

}

private static void passToken() {

int currentIndex = 0;

try {

while (true) {

// Send the token to the current client

PrintWriter out = new PrintWriter(clientSockets.get(currentIndex).getOutputStream(), true);

out.println("TOKEN");

System.out.println("Token passed to Client " + (currentIndex + 1));

// Wait for acknowledgment

BufferedReader in = new BufferedReader(new InputStreamReader(clientSockets.get(currentIndex).getInputStream()));

in.readLine(); // Wait for the acknowledgment from client

// Pass the token to the next client in the ring

currentIndex = (currentIndex + 1) % clientCount;

Thread.sleep(1000); // Sleep for a while before passing the token again

}

} catch (IOException | InterruptedException e) {

e.printStackTrace();

}

}

}

**2. TokenRingClient.java**

The client receives the token from the server and enters its critical section.

java

CopyEdit

import java.io.\*;

import java.net.\*;

public class TokenRingClient {

private static final String SERVER\_ADDRESS = "127.0.0.1"; // Change to server IP for 2 physical machines

private static final int PORT = 12345;

public static void main(String[] args) {

try {

Socket socket = new Socket(SERVER\_ADDRESS, PORT);

System.out.println("Connected to Token Ring Server.");

// Listen for token from the server

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

while (true) {

String message = in.readLine();

if ("TOKEN".equals(message)) {

// Enter critical section

System.out.println("Token received. Entering Critical Section...");

// Simulate work in critical section

Thread.sleep(2000); // Simulate critical section processing

// Exit critical section

System.out.println("Exiting Critical Section.");

// Send acknowledgment to server

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

out.println("ACK");

}

}

} catch (IOException | InterruptedException e) {

e.printStackTrace();

}

}

}