**TokenRingServer.java (Server that manages the token)**

This server simulates the ring of processes and ensures token passing.

java

CopyEdit

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class TokenRingServer {

private static final int PORT = 12345;

private static final List<ProcessClient> clients = new ArrayList<>();

private static int tokenHolderID = -1; // Initially, no token holder.

public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket(PORT);

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

// Accept clients (processes)

while (true) {

Socket clientSocket = serverSocket.accept();

ProcessClient client = new ProcessClient(clientSocket);

clients.add(client);

new Thread(client).start(); // Start each client in a new thread

}

} catch (IOException e) {

e.printStackTrace();

}

}

// Pass token to the next client in the ring

public static synchronized void passToken(int currentID) {

int nextID = (currentID + 1) % clients.size();

ProcessClient nextClient = clients.get(nextID);

if (tokenHolderID == -1) {

// First time, assign the token to the first client

tokenHolderID = 0;

System.out.println("Token given to Process " + (tokenHolderID + 1));

nextClient.receiveToken();

} else {

System.out.println("Passing token from Process " + (currentID + 1) + " to Process " + (nextID + 1));

nextClient.receiveToken();

}

}

public static void setLeader(int processID) {

System.out.println("Leader is Process " + (processID + 1));

}

}

**2. ProcessClient.java (Represents each client in the system)**

This client simulates a process that can receive the token and pass it.

java

CopyEdit

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class ProcessClient implements Runnable {

private Socket socket;

private int processID;

private TokenRingServer server;

public ProcessClient(Socket socket) {

this.socket = socket;

this.server = new TokenRingServer();

}

@Override

public void run() {

try {

DataInputStream inputStream = new DataInputStream(socket.getInputStream());

processID = inputStream.readInt();

System.out.println("Process " + (processID + 1) + " connected to the server.");

// Wait for token passing

while (true) {

// If it's this process' turn to have the token

if (processID == TokenRingServer.tokenHolderID) {

// If the process has the token, perform the action and pass the token

System.out.println("Process " + (processID + 1) + " has the token.");

// Mark itself as the leader if it's the first to start

if (TokenRingServer.tokenHolderID == 0) {

server.setLeader(processID);

}

// Simulate the token passing

server.passToken(processID);

break;

}

Thread.sleep(1000); // Wait for a while before checking again

}

} catch (IOException | InterruptedException e) {

e.printStackTrace();

}

}

// When a process receives the token

public void receiveToken() {

try {

DataOutputStream outputStream = new DataOutputStream(socket.getOutputStream());

outputStream.writeInt(processID);

System.out.println("Token received by Process " + (processID + 1));

System.out.println("Passing token to next process...");

server.passToken(processID);

} catch (IOException e) {

e.printStackTrace();

}

}

}