**BerkeleyMaster.java**

java

CopyEdit

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class BerkeleyMaster {

private static final int PORT = 12345;

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

private static int clientCount = 2; // assuming there are 2 clients

public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket(PORT);

System.out.println("Master Server is running...");

// Accept clients

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

Socket clientSocket = serverSocket.accept();

clientSockets.add(clientSocket);

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

}

// Get current time from the master

long masterTime = System.currentTimeMillis();

System.out.println("Master time: " + masterTime);

// Send the master time to clients

for (Socket clientSocket : clientSockets) {

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

out.println(masterTime);

}

// Receive time differences from clients

long totalDiff = 0;

for (Socket clientSocket : clientSockets) {

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

long clientTime = Long.parseLong(in.readLine());

long diff = clientTime - masterTime;

totalDiff += diff;

System.out.println("Client time: " + clientTime + " Difference: " + diff);

}

// Calculate the average offset

long averageOffset = totalDiff / clientCount;

System.out.println("Average offset: " + averageOffset);

// Send the average offset to clients

for (Socket clientSocket : clientSockets) {

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

out.println(averageOffset);

}

// Close connections

for (Socket clientSocket : clientSockets) {

clientSocket.close();

}

serverSocket.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

#### BerkeleyClient.java

java

CopyEdit

import java.io.\*;

import java.net.\*;

public class BerkeleyClient {

private static final int PORT = 12345;

private static String masterAddress = "127.0.0.1"; // IP address of master server (localhost for now)

public static void main(String[] args) {

try {

Socket socket = new Socket(masterAddress, PORT);

System.out.println("Connected to master server.");

// Get the time from master server

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

long masterTime = Long.parseLong(in.readLine());

System.out.println("Master time: " + masterTime);

// Get the local time of this client

long clientTime = System.currentTimeMillis();

System.out.println("Client local time: " + clientTime);

// Calculate the time difference

long diff = clientTime - masterTime;

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

out.println(clientTime); // Send client time to master

// Receive the average offset from the master

long averageOffset = Long.parseLong(in.readLine());

System.out.println("Average offset from master: " + averageOffset);

// Adjust the clock

long adjustedTime = clientTime - diff + averageOffset;

System.out.println("Adjusted client time: " + adjustedTime);

socket.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}