ryt current

import java.io.\*;

import java.net.\*;

import java.lang.String;

import global.\*;

class ClientWorker implements Runnable {

private ServerSocket serv;

private Socket client;

private static int num = 0 ;

private int id;

ClientWorker(ServerSocket serv) {

this.serv = serv;

num++;

this.id = num;

}

public void run(){

String line;

BufferedReader in = null;

PrintWriter out = null;

String clientSentence ,modifiedSentence ;

String capitalizedSentence;

try

{

while(true) {

//System.out.println(" In " +g.i);

client = serv.accept();

//System.out.println(" accepted " +g.i);

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(client.getInputStream()));

//System.out.println("buferread " +g.i);

DataOutputStream outToClient = new DataOutputStream(client.getOutputStream());

//System.out.println(" dataout " +g.i);

clientSentence = inFromClient.readLine();

// System.out.println("Client sent: " + clientSentence + " id is " + id);

// num = Integer.parseInt(clientSentence);

if (! (InetAddress.getByName(g.ip[g.i])).isReachable(4440) )

{

while(!(InetAddress.getByName(g.ip[g.i])).isReachable(4440) )

{

g.flag[g.i] = false ; g.i++; g.i = g.i%3;

}

} else g.flag[g.i] = true;

//capitalizedSentence = clientSentence.toUpperCase() + '\n';

Socket clientSocket = new Socket(g.ip[g.i], 4444);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

// Create (buffered) input stream attached to socket

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

// Write line to server

outToServer.writeBytes(clientSentence + '\n');

modifiedSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + modifiedSentence);

outToClient.writeBytes(modifiedSentence);

client.close(); // System.out.println(" I " +g.i);

if (! (InetAddress.getByName(g.ip[g.i])).isReachable(4440)){ g.flag[g.i]= false; break;}

// System.out.println ("jerk");

}

}

catch (IOException e) {e.printStackTrace(); }

}

}

public class ryt {

public static void main(String[] args) throws Exception {

//try {

ServerSocket welcomeSocket = new ServerSocket(4444);

/\*} catch (IOException e) {

System.out.println("No I/O");

System.exit(1);

}\*/ //boolean k = true;

// global g;

//ServerSocket echoServer = new ServerSocket(6789);

//accept requests

// declare an adjacency matrix for g.ip addr and g.flag for if\_not\_down

long start = System.currentTimeMillis();

long end = start + 1\*1000; // 6 seconds \* 1000 ms/secd

while(true)

{

if(System.currentTimeMillis() == end)

{

// System.out.println("out I " +g.i);

g.i++;

g.i = g.i%3;

//System.out.println(" I " +g.i);

ClientWorker w; // = new ClientWorker(welcomeSocket);

try{

// if\_not\_down(g.i) getByName(g.ip[g.i])

InetAddress adr = InetAddress.getByName("0.0.0.0");

// if(g.flag[g.i])

{

adr = InetAddress.getByName(g.ip[g.i]); // for those g.ip addr which are not down

System.out.println("Host" + g.ip[g.i] +" is reachable: " +adr.isReachable(4440));

// System.out.println(" I " +g.i);

}

if (!adr.isReachable(4440))

{

g.flag[g.i] = false;

}

else

{ g.flag[g.i] = true;// System.out.println("hell");

w = new ClientWorker(welcomeSocket);

Thread t = new Thread(w);// System.out.println("hell1");

t.start(); //System.out.println("hell2");

}

} catch (IOException e) {

e.printStackTrace();

}

//forward requests if reachable and set unreachable to false

end = System.currentTimeMillis() + 1\*1000;

}

if(System.currentTimeMillis()> end) break;

}

}

}

ryt: v3 15th feb

import java.io.\*;

import java.net.\*;

import java.lang.String;

import global.\*;

class ClientWorker implements Runnable {

private ServerSocket serv;

private Socket client;

private static int num = 0 ;

private int id;

ClientWorker(ServerSocket serv) {

this.serv = serv;

num++;

this.id = num;

}

public void run(){

String line;

BufferedReader in = null;

PrintWriter out = null;

int j= g.i ;

String clientSentence ,modifiedSentence ;

try

{

while(true) {

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(client.getInputStream()));

DataOutputStream outToClient = new DataOutputStream(client.getOutputStream());

clientSentence = inFromClient.readLine();

System.out.println("Client sent: " + clientSentence + " id is " + id);

if (! (InetAddress.getByName(g.ip[j])).isReachable(4440) )

{

while((!(InetAddress.getByName(g.ip[j])).isReachable(4440)) && ( g.flag[j] !=2))

{

g.flag[j] = 3 ; j++; j = j%3;

}

} else g.flag[j] = 1;

Socket clientSocket = new Socket(g.ip[j], 4444);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

// Create (buffered) input stream attached to socket

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

// Write line to server

outToServer.writeBytes(clientSentence + '\n');

modifiedSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + modifiedSentence);

outToClient.writeBytes(modifiedSentence);

client.close(); System.out.println(" I " +j);

if (! (InetAddress.getByName(g.ip[j])).isReachable(4440)){ g.flag[j]= 2; System.out.println (" breaking "); break;}

System.out.println ("jerk");

continue;

}

}

catch (IOException e) {e.printStackTrace(); }

}

}

public class ryt {

public static void main(String[] args) throws Exception {

//try {

ServerSocket welcomeSocket = new ServerSocket(4444);

Socket pumba;

/\*} catch (IOException e) {

System.out.println("No I/O");

System.exit(1);

}\*/

long start = System.currentTimeMillis();

long end = start + 1\*1000; // 6 seconds \* 1000 ms/sec

while(true)

{

if(System.currentTimeMillis() == end)

{

//System.out.println("out I " +g.i);

g.i++;

g.i = g.i%3;

//System.out.println(" I " +g.i);

ClientWorker w ;

try{

InetAddress adr = InetAddress.getByName("0.0.0.0");

{

adr = InetAddress.getByName(g.ip[g.i]); // for those g.ip addr which are not down

if (g.flag[g.i] == 1)

System.out.println("Host" + g.ip[g.i] +" is reachable " );// +adr.isReachable(4440));

else if (g.flag[g.i] == 2) System.out.println("Host" + g.ip[g.i] +" busy " );

else System.out.println("Host" + g.ip[g.i] +" unreachable " );

}

if (!adr.isReachable(4440))

{

g.flag[g.i] = 3;

}

else if( g.flag[g.i] !=2)

{ //g.flag[g.i] = 1;// System.out.println("hell");

pumba = welcomeSocket.accept();

//if (pumba !=null)

{

while (g.flag[g.i] != 1 ) g.i++;

w = new ClientWorker(welcomeSocket);

Thread t = new Thread(w); System.out.println("hell1");

t.start(); //System.out.println("hell2");

g.flag[g.i] = 2;

}

}

} catch (IOException e) {

e.printStackTrace();

}

//forward requests if reachable and set unreachable to false

end = System.currentTimeMillis() + 1\*1000;

}

if(System.currentTimeMillis()> end) break;

}

}

}

ryt : v2 threads

import java.io.\*;

import java.net.\*;

import java.lang.String;

import global.\*;

class ClientWorker implements Runnable {

private ServerSocket serv;

private Socket client;

private static int num = 0 ;

private int id;

ClientWorker(ServerSocket serv) {

this.serv = serv;

num++;

this.id = num;

}

public void run(){

String line;

BufferedReader in = null;

PrintWriter out = null;

String clientSentence ,modifiedSentence ;

String capitalizedSentence;

try

{

while(true) {

//System.out.println(" In " +g.i);

client = serv.accept();

//System.out.println(" accepted " +g.i);

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(client.getInputStream()));

//System.out.println("buferread " +g.i);

DataOutputStream outToClient = new DataOutputStream(client.getOutputStream());

//System.out.println(" dataout " +g.i);

clientSentence = inFromClient.readLine();

System.out.println("Client sent: " + clientSentence + " id is " + id);

// num = Integer.parseInt(clientSentence);

if (! (InetAddress.getByName(g.ip[g.i])).isReachable(4440) )

{

while(!(InetAddress.getByName(g.ip[g.i])).isReachable(4440) )

{

g.flag[g.i] = false ; g.i++; g.i = g.i%3;

}

} else g.flag[g.i] = true;

//capitalizedSentence = clientSentence.toUpperCase() + '\n';

Socket clientSocket = new Socket(g.ip[g.i], 4444);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

// Create (buffered) input stream attached to socket

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

// Write line to server

outToServer.writeBytes(clientSentence + '\n');

modifiedSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + modifiedSentence);

outToClient.writeBytes(modifiedSentence);

client.close(); System.out.println(" I " +g.i);

if (! (InetAddress.getByName(g.ip[g.i])).isReachable(4440)){ g.flag[g.i]= false; break;}

System.out.println ("jerk");

}

}

catch (IOException e) {e.printStackTrace(); }

}

}

public class ryt {

public static void main(String[] args) throws Exception {

//try {

ServerSocket welcomeSocket = new ServerSocket(4444);

/\*} catch (IOException e) {

System.out.println("No I/O");

System.exit(1);

}\*/ //boolean k = true;

// global g;

//ServerSocket echoServer = new ServerSocket(6789);

//accept requests

// declare an adjacency matrix for g.ip addr and g.flag for if\_not\_down

long start = System.currentTimeMillis();

long end = start + 1\*1000; // 6 seconds \* 1000 ms/secd

while(true)

{

if(System.currentTimeMillis() == end)

{

System.out.println("out I " +g.i);

g.i++;

g.i = g.i%3;

System.out.println(" I " +g.i);

ClientWorker w ; //= new ClientWorker(wel);

try{

// if\_not\_down(g.i) getByName(g.ip[g.i])

InetAddress adr = InetAddress.getByName("0.0.0.0");

// if(g.flag[g.i])

{

adr = InetAddress.getByName(g.ip[g.i]); // for those g.ip addr which are not down

System.out.println("Host" + g.ip[g.i] +" is reachable: " +adr.isReachable(4440));

System.out.println(" I " +g.i);

}

if (!adr.isReachable(4440))

{

g.flag[g.i] = false;

}

else

{ g.flag[g.i] = true;// System.out.println("hell");

w = new ClientWorker(welcomeSocket);

Thread t = new Thread(w);// System.out.println("hell1");

t.start(); //System.out.println("hell2");

}

} catch (IOException e) {

e.printStackTrace();

}

//forward requests if reachable and set unreachable to false

end = System.currentTimeMillis() + 1\*1000;

}

if(System.currentTimeMillis()> end) break;

}

}

}

ryt v1.0

import java.io.\*;

import java.net.\*;

import java.lang.String;

public class ryt {

public static void main(String[] args) throws Exception {

String clientSentence ,modifiedSentence ;

String capitalizedSentence;

//try {

ServerSocket welcomeSocket = new ServerSocket(4444);

/\*} catch (IOException e) {

System.out.println("No I/O");

System.exit(1);

}\*/

String[] ip = {"192.168.18.71","192.168.18.50","192.168.18.46"};

boolean[] flag = {true,true,true};

int i = -1;

//accept requests

// declare an adjacency matrix for ip addr and flag for if\_not\_down

long start = System.currentTimeMillis();

long end = start + 1\*1000; // 6 seconds \* 1000 ms/sec

while(true)

{

if(System.currentTimeMillis() == end)

{

System.out.println("out I " +i);

i++;

i = i%3;

System.out.println(" I " +i);

try

{ // if\_not\_down(i) getByName(ip[i])

InetAddress adr = InetAddress.getByName("0.0.0.0");

// if(flag[i])

{

adr = InetAddress.getByName(ip[i]); // for those ip addr which are not down

System.out.println("Host" + ip[i] +" is reachable: " +adr.isReachable(4440));

System.out.println(" I " +i);

}

if (!adr.isReachable(4440))

{

flag[i] = false;

}

else

{ flag[i] = true;

while(true) {

System.out.println(" In " +i);

Socket connectionSocket = welcomeSocket.accept(); System.out.println(" accepted " +i);

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

System.out.println("buferread " +i);

DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

System.out.println(" dataout " +i);

clientSentence = inFromClient.readLine();

System.out.println("Client sent: " + clientSentence);

if (! (InetAddress.getByName(ip[i])).isReachable(4440) )

{

while(!(InetAddress.getByName(ip[i])).isReachable(4440) )

{

flag[i] = false ; i++; i = i%3;

}

} else flag[i] = true;

//capitalizedSentence = clientSentence.toUpperCase() + '\n';

Socket clientSocket = new Socket(ip[i], 4444);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

// Create (buffered) input stream attached to socket

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

// Write line to server

outToServer.writeBytes(clientSentence + '\n');

modifiedSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + modifiedSentence);

outToClient.writeBytes(modifiedSentence);

connectionSocket.close(); System.out.println(" I " +i);

if (! (InetAddress.getByName(ip[i])).isReachable(4440)){ flag[i]= false; break;}

System.out.println ("jerk");

}

}

} catch (IOException e) {e.printStackTrace(); }

//forward requests if reachable and set unreachable to false

end = System.currentTimeMillis() + 1\*1000;

}

if(System.currentTimeMillis()> end) break;

}

}

}

import java.io.\*;

import java.net.\*;

import java.lang.String;

public class ryt {

public static void main(String[] args) throws Exception {

String clientSentence ,modifiedSentence ;

String capitalizedSentence;

//try {

ServerSocket welcomeSocket = new ServerSocket(4444);

/\*} catch (IOException e) {

System.out.println("No I/O");

System.exit(1);

}\*/

String[] ip = {"192.168.18.71","192.168.18.141","192.168.18.86"};

boolean[] flag = {true,true,true};

int i = -1;

//accept requests

// declare an adjacency matrix for ip addr and flag for if\_not\_down

long start = System.currentTimeMillis();

long end = start + 1\*1000; // 6 seconds \* 1000 ms/sec

while(true)

{

if(System.currentTimeMillis() == end)

{ System.out.println("out I " +i);

i++;

i = i%3; System.out.println(" I " +i);

try

{ // if\_not\_down(i) getByName(ip[i])

InetAddress adr = InetAddress.getByName("0.0.0.0");

// if(flag[i])

{

adr = InetAddress.getByName(ip[i]); // for those ip addr which are not down

System.out.println("Host" + ip[i] +" is reachable: " +adr.isReachable(4440));

System.out.println(" I " +i);

}

if (!adr.isReachable(4440))

{

flag[i] = false;

}

else

{ flag[i] = true;

while(true) { System.out.println(" In " +i);

Socket connectionSocket = welcomeSocket.accept(); System.out.println(" accepted " +i);

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

System.out.println("buferread " +i);

DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

System.out.println(" dataout " +i);

clientSentence = inFromClient.readLine();

System.out.println("Client sent: " + clientSentence);

if (! (InetAddress.getByName(ip[i])).isReachable(4440) )

{ while(!(InetAddress.getByName(ip[i])).isReachable(4440) )

{flag[i] = false ; i++; i = i%3;}

} else flag[i] = true;

//capitalizedSentence = clientSentence.toUpperCase() + '\n';

Socket clientSocket = new Socket(ip[i], 4444);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

// Create (buffered) input stream attached to socket

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

// Write line to server

outToServer.writeBytes(clientSentence + '\n');

modifiedSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + modifiedSentence);

outToClient.writeBytes(modifiedSentence);

connectionSocket.close(); System.out.println(" I " +i);

if (! (InetAddress.getByName(ip[i])).isReachable(4440)){ flag[i]= false; break;}

System.out.println ("jerk");

}

}

} catch (IOException e) {e.printStackTrace(); }

//forward requests if reachable and set unreachable to false

end = System.currentTimeMillis() + 1\*1000;

}

if(System.currentTimeMillis()> end) break;

}

}

}