Laboratorul 3 – Sisteme distribuite

Ne propunem să identificăm oprirea funcționării (defectarea) unui proces de către alt proces.

> *".. a heartbeat is a periodic signal generated by hardware or software to indicate normal operation or to synchronize other parts of a computer system. Usually a heartbeat is sent between machines at a regular interval in the order of seconds. If the endpoint does not receive a heartbeat for a time—usually a few heartbeat intervals—the machine that should have sent the heartbeat is assumed to have failed."*
>
> https://en.wikipedia.org/wiki/Heartbeat_(computing)

TEMA DE LUCRU: Implementați un mecanism de tip heartbeat (java, consola, linux); rulați 3 astfel de procese în rețeaua locală care se verifică periodic unul pe altul.

In acest scenariu fiecare proces lansat are informații despre celelalte procese (ip,port).

Pentru simplificare am optat pentru un singur cod sursă (portul este fixat), iar procesele sunt lansate din mașini diferite (ip diferite) in rețeaua locală - similar se pot rula 3 procese pe aceeasi mașină, dar cu porturi distincte.

Adresa IP poate fi identificată la pornirea programului:

```
static String my_ip;

// se identifica adresa IP
try(final DatagramSocket socket = new DatagramSocket()){
	socket.connect(InetAddress.getByName("8.8.8.8"), 10002);
	my_ip = socket.getLocalAddress().getHostAddress();
}
```

Obs: atenție la sistemele cu mai multe adrese IP (reale, virtuale).

Fiecare proces știe de existența celorlalte (nume, id, ip):

```
// vezi si configurare  /etc/hosts

static HashMap<Integer,String> processes= new HashMap<Integer,String>();

processes.put(1, "node01");            // sau 192.168.199.251
processes.put(2, "node02");            // sau 192.168.199.252
processes.put(3, "node03");            // sau 192.168.199.253
```

Fiecare proces se poate identifica ("afla cine este") in funcție de adresa IP:

```
static int self_id=-1;                 // urmeaza identificarea

InetAddress iAddress = InetAddress.getLocalHost();
String my_name = iAddress.getHostName();

if(my_name != null && !my_name.isEmpty()){
       if(my_name.equals("node01")){                 //sau 192.168.199.251
                     self_id = 1;
       }
       else if( ...
...
```

Programul va conține un fir de execuție pentru recepționarea mesajelor (și transmiterea confirmării) și un fir de execuție pentru transmiterea periodică a mesajelor de testare (heartbeat) către celelalte procese.

De exemplu:

```
public void run(){

       if(operation.equals("receiver")){
          ServerSocket serverSocket = null;
          try{
              serverSocket = new ServerSocket(server_Port);
              while(true){
                 Socket socket = serverSocket.accept();
                 // System.out.println("Connection established.....");
                 DataInputStream in = new DataInputStream(socket.getInputStream());
                 String option=in.readUTF();
                 if(option.equals("heartbeat")){
                   int sender=Integer.parseInt(in.readUTF());
                   System.out.println("HEARTBEAT received from " + processes.get(sender));
                 }
                 socket.close();
              } // while
          }
          catch(Exception e){
              e.printStackTrace();
          }
       }
```

```
	else if(operation.equals("heartbeat")){
		while(true) {
		   for (int key : processes.keySet())
		      if(key!=self_id) {
			 String destination_server=processes.get(key);
			 try{
				Thread.sleep(1250);
				System.out.println("try to check " + destination_server);
				Socket socket = new Socket(destination_server, server_Port);
				DataOutputStream out =
					new DataOutputStream(socket.getOutputStream());
				out.writeUTF("heartbeat");
				out.writeUTF(self_id+"");
				System.out.println("Sent HEARTBEAT to: "+destination_server);
			 }
			 catch(Exception e){
				System.out.println("\n***\tpeer has FAILED!\t***");
			 }
		      }
		}
	}
 }
```

Exemplificare clasa HeartBeat:

```
public class HeartBeat implements Runnable {
	static int self_id=-1;
	static int server_Port = 5533;
	String operation;                       // tip fir RECEIVER, HEARTBEAT
	static HashMap<Integer,String> processes= new HashMap<Integer,String>();
	static boolean received=false;
	static String my_ip;

	public HeartBeat(String operation) {
		this.operation = operation;
	}

	/* The main() method starts two threads, RECEIVER and HEARTBEAT */
	public static void main(String args[])
		throws UnknownHostException, IOException, InterruptedException {

	Thread.sleep(100);
	initialize();

	Runnable receiver = new HeartBeat("receiver");
	new Thread(receiver).start();

	Runnable heartbeat = new HeartBeat("heartbeat");
	new Thread(heartbeat).start();

	while(true){}                           // bucla principala
}
```

Exemplu rulare:

```
ccostea@node02:~$ java HeartBeat

initializare nod 192.168.199.252
my InetAddress is node02/127.0.1.1
my server_name is node02
my ip is 192.168.199.252
self_id = 2

Receiver started..

Heartbeat started..

try to check node01
***     peer has FAILED!        ***
try to check node03
***     peer has FAILED!        ***
try to check node01
***     peer has FAILED!        ***
try to check node03
***     peer has FAILED!        ***
try to check node01
***     peer has FAILED!        ***
try to check node03
***     peer has FAILED!        ***
try to check node01
***     peer has FAILED!        ***
try to check node03
***     peer has FAILED!        ***
Connection established.....
HEARTBEAT received from node01
try to check node01
Sent HEARTBEAT to: node01
try to check node03
***     peer has FAILED!        ***
Connection established.....
HEARTBEAT received from node01
try to check node01
Sent HEARTBEAT to: node01
try to check node03
***     peer has FAILED!        ***
Connection established.....
HEARTBEAT received from node01
try to check node01
Sent HEARTBEAT to: node01
try to check node03
***     peer has FAILED!        ***
Connection established.....
HEARTBEAT received from node01
try to check node01
Sent HEARTBEAT to: node01
try to check node03
```

```
***     peer has FAILED!        ***
Connection established.....
HEARTBEAT received from node01
try to check node01
Sent HEARTBEAT to: node01
try to check node03
***     peer has FAILED!        ***
Connection established.....
HEARTBEAT received from node01
try to check node01
Sent HEARTBEAT to: node01
try to check node03
***     peer has FAILED!        ***
Connection established.....
HEARTBEAT received from node01
try to check node01
Sent HEARTBEAT to: node01
try to check node03
***     peer has FAILED!        ***
try to check node01
***     peer has FAILED!        ***
try to check node03
***     peer has FAILED!        ***
try to check node01
***     peer has FAILED!        ***
try to check node03
***     peer has FAILED!        ***
^Cccostea@node02:~$
```