

TSR: First Partial

This exam consists of 20 multiple choice questions. In every case only one answer is correct. You should answer in a separate sheet. If correctly answered, they contribute 0,5 points to the exam grade. If incorrectly answered, the contribution is negative: -0.167. So, think carefully your answers.

THEORY

1. Regarding Unit 1, these sentences correctly describe some aspects of distributed systems:

a	Every concurrent system is a distributed system.
b	The e-mail service is an example of distributed system.
c	The agents in a distributed system cannot share any resource since each agent must be placed in a different computer.
d	The developer of a distributed application does not need to worry about fault tolerance, since it is inherently guaranteed by the distributed system.

2. One of the reasons for stating that Wikipedia is a scalable distributed application is...

a	Since its first release, it has been implemented following the SaaS cloud service model.
b	It is a LAMP system, and all systems of this kind are highly scalable.
c	It uses a P2P interaction approach and this strongly enhances its scalability.
d	It uses caching at its reverse proxies and component replication.

3. The main goal of the PaaS cloud service model is...

a	To automate the configuration, deployment and upgrading of distributed services and their reconfiguration under varying loads.
b	To automate infrastructure provision.
c	To provide distributed services under a pay-as-you-go model.
d	To provide persistent data under a pay-as-you-go model.

4. A simple distributed system model was proposed in Unit 2 because that model...

a	...ensures data persistence.
b	...is needed for comparing multi-threaded and asynchronous programming.
c	...facilitates a good basis for designing distributed algorithms and protocols and for reasoning about their correctness before starting their implementation.
d	...shows that activity blocking prevents services from scaling.

TSR

5. This is the best solution for ensuring data persistence:

a	Usage of stateful servers.
b	Data replication.
c	Use the most reliable hard disk drives.
d	Avoidance of concurrent accesses to data.

6. The simple system model described in Unit 2 is directly supported by the asynchronous programming paradigm because...

a	...asynchronous programming is based on causal communication.
b	...processes in that simple system model are multi-threaded.
c	...there is a direct translation between guards + actions in the model and events + callbacks in asynchronous programming.
d	...processes inherently follow the stop failure model in the asynchronous programming paradigm.

7. A messaging middleware layer (MML) is more convenient than remote method invocation (RMI) for building scalable applications because...

a	MML provides location transparency and RMI cannot do this.
b	MML is inherently asynchronous, while RMI is synchronous.
c	Processes that use MML assume a shared resource space. In RMI none of the processes shares any resource.
d	Processes that use MML are automatically replicated. In RMI, replication isn't allowed.

8. Persistent messaging...

a	...implies location transparency.
b	...is automatically guaranteed when a naming middleware is used.
c	...may be easily implemented by intermediate message brokers.
d	...cannot be used in asynchronous communication.

TSR

SEMINARS

9. Considering this program:

```
var fs=require('fs');
if (process.argv.length<5) {
    console.error('More file names are needed!!');
    process.exit();
}
var files = process.argv.slice(2);
var i=-1;
do {
    i++;
    fs.readFile(files[i], 'utf-8', function(err,data) {
        if (err) console.log(err);
        else console.log('File '+files[i]+' : '+data.length+' bytes. ');
    })
} while (i<files.length-1);
console.log('We have processed '+files.length+' files.');
```

These sentences are true if we assume that no error aborts this program execution:

a	This JavaScript program prints in all iterations, among other data, the name of the last file provided in the command line.
b	It prints the name and length for each one of the files received from the command line.
c	It prints “We have processed N files” at the end of its execution, being N the number of file names given as arguments.
d	It discards the names of the first two files given as arguments to this program.

10. Regarding the program shown in the previous question...

a	It needs multiple turns for completing its execution, since each file being read requires its own turn.
b	It generates an exception and crashes if any error happens when it tries to read a file.
c	This program is incorrect. It must use “var i=0” to initialise variable “i” in order to be correct.
d	It always prints the same length in all iterations. We need a closure in order to avoid this faulty behaviour.

11. Regarding the mutual exclusion algorithms seen in Seminar 2, it is true that...

a	The central server algorithm minimises the amount of messages being needed.
b	The virtual unidirectional ring algorithm has a synchronisation delay of 1 message.
c	The synchronisation delay of the multicast algorithm with logical clocks is $2N-2$ messages.
d	The multicast algorithm based on quorums (i.e., its version described in the slides) complies with all 3 mutual exclusion correctness conditions.

TSR

12. Considering this program...

```
var ev = require('events');
var emitter = new ev.EventEmitter;
var num1 = 0;
var num2 = 0;
function emit_e1() { emitter.emit("e1") }
function emit_e2() { emitter.emit("e2") }
emitter.on("e1", function() {
  console.log( "Event e1 has happened " + ++num1 + " times.");});
emitter.on("e2", function() {
  console.log( "Event e2 has happened " + ++num2 + " times.");});
emitter.on("e1", function() {
  setTimeout( emit_e2, 3000 )});
emitter.on("e2", function() {
  setTimeout( emit_e2, 2000 )});
setTimeout( emit_e1, 2000 );
```

The following sentences are true:

a	Event “e1” happens only once, 2 seconds after this program is started.
b	Event “e2” never happens.
c	The period of “e2” is five seconds.
d	The period of “e1” is three seconds.

13. Considering the program shown in the previous question...

a	The first “e2” event happens five seconds after the program is started.
b	No event is generated in its execution, since its emit() calls are incorrect.
c	We cannot have more than one listener for each event. Therefore, the program is aborted by an exception in its third emitter.on() call.
d	None of its events happens periodically.

14. The ØMQ REQ-REP communication pattern is considered synchronous because...

a	It follows the client/server interaction pattern and in that pattern the client remains blocked until a reply is received.
b	Both REQ and REP sockets are bidirectional; i.e., both may send and receive messages.
c	The output queue in the REQ socket has a limited capacity. It may only hold one message.
d	REQ sockets cannot transmit a request until the reply to its previous request is received. REP sockets cannot forward a reply until its request is received.

TSR

15. Considering these two node.js programs...

<pre>// server.js var net = require('net'); var server = net.createServer(function(c) { // 'connection' listener console.log('server connected'); c.on('end', function() { console.log('server disconnected'); }); c.on('data', function(data) { console.log('Request: ' + data); c.write(data+ 'World!'); }); }); server.listen(9000);</pre>	<pre>// client.js var net = require('net'); var i=0; var client = net.connect({port: 9000}, function() { client.write('Hello '); }); client.on('data', function(data) { console.log('Reply: ' + data); i++; if (i==2) client.end(); }); client.on('end', function() { console.log('client ' + 'disconnected'); });</pre>
--	--

The following sentences are true:

a	The server terminates after sending its first reply to the first client.
b	The client never terminates.
c	This server can only handle one connection.
d	This client cannot connect to this server.

16. Leader election algorithms (from Seminar 2)...

a	...have no safety condition.
b	...may be infinitely looking for a leader process.
c	...must ensure that a single leader is chosen.
d	...must respect causal order.

17. We want to implement a mutual exclusion service using NodeJS and ØMQ, using the first algorithm explained in Seminar 2: the central server algorithm. In order to implement this service, the best of the following options is...

a	The server needs a DEALER and a ROUTER socket to balance the load among its clients.
b	Each client needs a DEALER socket to interact with the server.
c	Each client needs a REP socket to interact with the server.
d	Each client needs a SUB socket to interact with the server.

TSR

18. We want to implement a mutual exclusion service using NodeJS and ØMQ, using the second algorithm explained in Seminar 2: the (virtual) unidirectional ring algorithm. In order to implement this service...:

a	We need to use any leader election algorithm in order to choose a coordinator process.
b	All processes have the same role and need a REP socket to send messages and a REQ socket to receive them.
c	All processes have the same role and need a PUSH socket to send the token and a PULL socket to receive it.
d	All processes have the same role and use a PUB socket to send the token and a DEALER socket to receive it.

19. Considering these programs...

<pre>//client.js var zmq=require('zmq'); var rq=zmq.socket('dealer'); rq.connect('tcp://127.0.0.1:8888'); for (var i=1; i<100; i++) { rq.send(''+i); console.log("Sending %d",i); } rq.on('message',function(req,rep){ console.log("%s %s",req,rep); });</pre>	<pre>// server.js var zmq = require('zmq'); var rp = zmq.socket('dealer'); rp.bindSync('tcp://127.0.0.1:8888'); rp.on('message', function(msg) { var j = parseInt(msg); rp.send([msg, (j*3).toString()]); });</pre>
---	---

The following sentences are true:

a	Both client and server exchange messages in a synchronous way in this example, since they follow a request-reply pattern.
b	The server returns a message with 2 segments to the client. The second segment contains a value that is 3 times greater than that in the first segment.
c	Client and server may be run in different computers. They interact without problems in that case.
d	No message is sent by the client since the '' + i statement generates an exception and the program aborts at that point.

20. Please consider which of the following variations will generate new programs with the same behaviour as that shown in question 19 (A --> B means that statement A must be replaced by statement B)...

a	The 'rq' socket should be of type 'PULL' and the 'rp' of type 'PUSH'.
b	The 'rq' socket should be a 'PUSH' socket and 'rp' should be a 'PULL'.
c	Client: rq.connect('tcp://127.0.0.1:8888'); --> rq.bindSync('tcp://*:8888'); Server: rp.bindSync('tcp://127.0.0.1:8888'); --> rp.connect('tcp://127.0.0.1:8888');
d	The 'rq' socket should be of type 'REP' and 'rp' should be of type 'REQ'.