

TSR: Primer parcial

Aquest examen comprèn 10 preguntes d'opció múltiple. En cada cas només una resposta és correcta. Ha de respondre's en una fulla a part. Les respostes correctes aporten 1 punt a la qualificació d'aquesta prova. Les incorrectes redueixen la qualificació 0.33 punts.

1. Quina afirmació sobre els sistemes LAMP és correcta?

A	Els components en un sistema LAMP es despleguen habitualment sobre nodes amb sistema Windows.
B	Un sistema LAMP sol tenir una arquitectura de 3 nivells.
C	Cap component en un sistema LAMP pot replicar-se.
D	Un sistema LAMP complet no pot desplegar-se sobre un únic node.

2. Una diferència entre la computació en el núvol (CN) i els clústers d'alta disponibilitat (AD) és:

A	Els clústers AD repliquen els components dels serveis, però els serveis desplegats en CN no usen replicació.
B	Els clústers AD s'utilitzen principalment per a aplicacions de ciència de dades; la CN se sol utilitzar per a desplegar serveis distribuïts.
C	Per a l'usuari, la CN és facilitada per proveïdors externs, mentre els clústers AD són comprats i mantinguts per l'empresa que els utilitza.
D	Els clústers AD solen utilitzar un model de servei IaaS i els sistemes de CN proporcionen un model de servei SaaS.

3. Un dels problemes fonamentals que han de resoldre els sistemes distribuïts és la gestió de defectes i fallades. Per a fer això, cal...

A	Enviar els missatges una sola vegada. D'una altra manera, aquells agents que vulguen atacar el nostre sistema arreplegarien informació rellevant sobre els nostres serveis.
B	Evitar mecanismes de detecció de fallades perquè aquests mecanismes necessiten molts recursos i la detecció no sol ser fiable.
C	Replicar els components dels serveis per a garantir que almenys una instància estiga disponible.
D	Totes les anteriors.

4. El principal objectiu del "middleware" és:

A	Millorar la seguretat del sistema.
B	Millorar l'escalabilitat del sistema.
C	Ocultar i resoldre múltiples problemes de comunicació que poden sorgir entre els components d'un servei.
D	Gestionar les fallades.

TSR

5. Considere aquest programa:

```
// Files.js
const fs = require('fs');
if (process.argv.length<3) {
  console.error('More file names are needed!!');
  process.exit();
}
function handler(name) {
  return function(err,data) {
    if (err) console.error(err);
    else console.log('File '+name+: '+data.length+' bytes. ');
  }
}
var files = process.argv.slice(2);
var i=-1;
do {
  i++;
  fs.readFile(files[i], 'utf-8', handler(files[i]))
} while (i<files.length-1);
console.log('We have processed '+files.length+' files.');
```

S'ha arribat a executar el programa amb aquesta ordre: “node Files A B”. A és un fitxer de text de 234781 bytes i B un altre fitxer de text amb 430 bytes. Quina és l'eixida mostrada en l'execució?

A	More file names are needed.
B	File A: 234781 bytes. File B: 430 bytes. We have processed 2 files.
C	“We have processed 2 files.” seguit per una línia per fitxer, mostrant el seu nom i grandària.
D	No arriba a mostrar-se res perquè el procés avorta en la seua línia 8 sense mostrar cap missatge.

6. En el programa de la qüestió anterior, la següent afirmació és certa:

A	Utilitza una promesa per a processar cada fitxer.
B	Avorta la seua execució en el cos de la funció “handler” perquè les funcions no poden retornar funcions.
C	Utilitza la funció fs.readFile de manera sincrònica per a evitar problemes quan els noms i grandàries dels fitxers hagen de mostrar-se.
D	La funció “handler” proporciona una clausura per a mostrar adequadament el nom de cada fitxer.

TSR

7. En l'algorisme d'exclusió mútua amb servidor central, és cert que:

A	És un algorisme altament disponible perquè tots els seus agents estan replicats per omissió.
B	Usa un patró sincrònic petició-resposta per a obtenir el permís d'accés a la secció crítica i un enviament unidireccional asincrònic per a alliberar-la.
C	Usa un patró unidireccional asincrònic PUSH-PULL per a obtenir el permís d'accés a la secció crítica i un PUB-SUB per a alliberar-la.
D	Usa un patró asincrònic PUB-SUB per a obtenir el permís d'accés a la secció crítica i un petició-resposta sincrònic per a alliberar-la.

8. Considerant aquests programes, a executar en un mateix ordinador...

<pre>// client.js var zmq=require('zmq'); var rq=zmq.socket('dealer'); rq.connect('tcp://127.0.0.1:8888'); var i=1; rq.send(''+i); rq.on('message',function(req,rep){ console.log("%s %s",req,rep); if (i==100) process.exit(1); rq.send(''+(++i)); });</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>
---	---

La següent afirmació és certa:

A	En el client, quan es mostra el valor del paràmetre “req”, el seu valor és idèntic al de la variable “i”.
B	El client envia 101 peticions al servidor abans d'acabar la seua execució.
C	No pot haver-hi dos o més instàncies del client en el mateix ordinador. Totes menys la primera avortarien en el seu intent de connectar amb el servidor.
D	Cap de les anteriors.

9. Considerant els programes mostrats en la qüestió anterior...

A	El client no pot enviar una nova petició fins que la resposta a l'anterior siga rebuda i processada.
B	Encara que el patró de comunicació DEALER-DEALER siga utilitzable, en aquest exemple els missatges no inclouen un delimitador. Sense ell, els missatges no s'entreguen.
C	Aquests programes són inútils perquè no es pot intercomunicar a dos processos utilitzant sockets DEALER en ambdós programes.
D	El servidor únicament pot processar les peticiones enviades per un sol client.

TSR

10. Supposem que una aplicació distribuïda necessita un canal de comunicació unidireccional asincrònic entre dos components A i B, enviant els missatges des d'A a B (és a dir, A --> B). Per a aconseguir aquesta comunicació, es pot utilitzar aquest patró ZeroMQ:

A	A: PULL, B: PUSH.
B	A: REQ, B: REP.
C	A: SUB, B: PUB.
D	A: DEALER, B: DEALER.