RETAKING EXAM 2nd BLOCK **Concurrency and Distributed Systems**
Units 6 to 9 - Practices 4, 5 and 6 **Date: 15th June 2018**

This exam has a maximum duration of 90 minutes.

This exam has a maximum score of **10 points**, equivalent to **3.5 points** of the final grade for the course. It contains questions of theoretical units and lab sessions. Indicate, for each of the following **58 statements**, if they are true (T) or false (F). **Each answer is worth: right= 10/58, wrong= -10/58, empty=0.**

Important: **first 3 errors do not penalize**, so they will be equivalent to an empty answer. From the 4th error (inclusive), the decrement for wrong answers will be applied.

THEORY

Given the set of tasks in a real-time system described by the following tables:

Task	T	Computation	Deadline	Priority
A	5	2	5	1
B	20	5	15	2
C	40	5	20	3
D	80	6	60	4

Task	Semaphore	Duration of the Critical Section
A	S1	2
B	S1 S2	1 2
C	S2 S3	2 2
D	S1 S3	1 3

Assuming an assignment of priorities in which the task with the lowest numerical value will have the highest priority, and considering that these tasks use the semaphores S1, S2 and S3 according to the previous table...

1. If these tasks do not use semaphores, then the system is schedulabled.	T
2. If the immediate priority ceiling protocol is applied, the response time of task C is 18.	F
3. If the immediate priority ceiling protocol is applied, the response time of task A is 3.	T
4. The blocking factor for task A is 1 and for task B is 2.	T

Regarding the priority inversion problem in a real-time system:

5. Given 2 tasks A, B, and A has higher priority than B. If B occupies a critical section CS and A executes the CS input protocol, the fact that A waits is called priority inversion.	T
6. When the priority inversion occurs, the waiting time is limited by the duration of the critical section in which the priority process waits.	F
7. To solve the priority inversion problem we must allow that the priority of a process can change temporarily.	T
8. Applying the immediate priority ceiling protocol, we guarantee the absence of deadlocks.	T

On the definition of Distributed Systems and their intrinsic characteristics:

9. Every concurrent system is a distributed system.	F
10. Every distributed system is implemented on a set of networked interconnected nodes.	T
11. Every real-time system is a distributed system.	F
12. Whatsapp is an example of a distributed system.	T
13. Distributed systems try to provide a single system image, that is, the user or client of the system observes it as if it was not distributed.	T

On scalability and security in distributed systems:

14. To improve scalability we must distribute the data, but centralize the responsibilities.	F
15. The use of caching does not improve the scalability of the systems, it only improves the performance in some cases.	F

Regarding transparency in distributed systems:

16. It usually implies a certain extra cost with respect to not providing such transparency.	T
17. In many cases it is impossible to achieve 100% transparency.	T
18. There are multiple axes of transparency. Some of them are: independence, remote access, ubiquity, efficiency.	F
19. When a distributed system hides in the identification of the resources where they are located, we say that it provides transparency of identification.	F
20. When a distributed system offers transparency of failures, users clearly perceive if a component of the system fails.	F

About standardization and design principles:

21. Modularity means splitting up the design into simpler components.	T
22. Standardization facilitates the usage of components (modules) developed by others.	T

Regarding the middleware concept:

23. The middleware is a software layer that facilitates the development of distributed applications.	T
24. The middleware determines the software architecture to be used in the development of distributed applications.	T

About the ROI communication mechanism:

25. The ORB is responsible for managing references to remote objects.	T
26. The format of the messages exchanged between client and server is transparent for the programmer.	T

About the Java RMI communication mechanism:

27. Remote objects are instances of classes that must implement an interface that extends Remote.	T
28. Java RMI is an object-oriented communication middleware that offers direct addressing and persistence.	F

Regarding the RESTful web services:

29. REST web services offer non-persistent communication and direct addressing through requests to the computer that hosts the service.	T
30. In REST, in the request, the type of http message (post, get, put, delete) indicates the type of operation on the resource (create, read, update, delete).	T

Regarding Java Message Service:

31. JMS avoids the need to include explicit sending / receiving operations in the code.	F
32. In JMS, destinations serve to communicate JMS Clients and JMS Providers with each other.	F

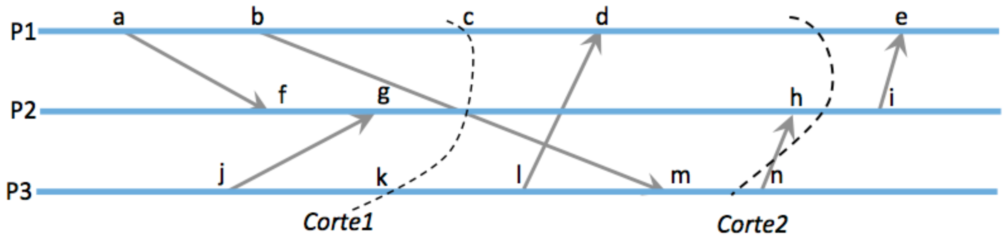
On the synchronization of physical clocks and the use of logical clocks:

33. In the Berkeley algorithm, the existence of a very precise clock is required with which to synchronize the clocks of the nodes of the system.	F
34. Since the Lamport logic clocks, together with the identifiers of the nodes, allow to establish a total order of the events, these can be used to untie between two TRY messages of the distributed mutual exclusion algorithm.	T
35. With logic clocks we can know the precise time in which the different events occur in a distributed system.	F
36. In Cristian's algorithm, if the server answers a client with a value equal to 1500, the new value of the client's local clock must be made equal to 1500 plus an adjustment value that will also be provided by the server.	F
37. Suppose we use the notation "Lamport logic clock . node identifier" and we know that the event "a" has the associated value 7.1 and the event "b" has the associated value 9.2, so we can affirm that the event "a" has occurred before event "b".	F
38. Given V (a) = [9, 5, 8, 7, 3] and V (b) = [3, 4, 1, 9, 10] two vector clocks, we can affirm that the events "a" and "b" are concurrent with each other.	T

About the consensus algorithm in the presence of failures seen in class:

39. It guarantees that the correct nodes decide the same value, even if half of the nodes fail.	F
40. Tolerates Byzantine failures and stop failures.	F
41. In case of multiple failures may require a high number of rounds.	V

Given the following schedule that shows the execution of three processes in a distributed system, each one in a different node:



42. The Lamport clocks of events "c" and "k" are zero, since they do not correspond to any event of sending or receiving messages.	F
43. It is observed that h-->n	F
44. Lamport clock for event "h" is C(h)=6	T
45. Vector clock for event "g" is V(g)=[1, 1, 1]	F
46. In this figure all events that have the same Lamport logic clock value are concurrent.	T
47. The Chandy-Lamport algorithm could have facilitated the cut indicated by Corte1, but not the one indicated by Corte2.	T

Regarding the algorithms of leader election and mutual exclusion in distributed systems seen in class:

48. In the leader election algorithm for rings, the initiator node finishes its work by broadcasting a COORDINATOR message through the ring indicating which is the new coordinator from now on.	T
49. In the centralized mutual exclusion algorithm, in order for a node to enter the critical section, first the rest of the nodes must notify the coordinator that they agree.	F
50. In the mutual exclusion algorithm for rings, when a node leaves the critical section, it returns the token to the coordinator node, so that it will eventually give it to the next node interested in entering the critical section.	F

LAB PRACTICE

Regarding practice 4 of Active Directory Domain Services (AD DS):

1. In this practice, to create a new domain, you needed to log in to the machine that was to be the controller of that domain.	T
2. After creating a new domain, setting up a machine (eg. adc1) as its domain controller, you need to use the local administrator account of adc1 to log in to adc1, and the system will ask you to change the password.	F
3. A user of a domain may be authorized to use resources from any other domain of the forest, except from the first domain of the forest.	F

Regarding practice 5 (The object-oriented distributed Chat based on RMI):

4. The ChatRobot process has a ChatUser object associated with it and receives the same messages as a ChatClient process.	T
5. The order of launching of the processes to start the chat distributed with a ChatRobot is the following: (1^{st}) rmiregistry, (2^{nd}) ChatRobot, (3^{rd}) ChatServer and (4^{th}) as many ChatClient as desired.	F
6. The objects ChatServer, ChatClient, ChatUser, ChatChannel and ChatMessage can be used remotely (proxies) because they implement an interface that inherits from Remote (java.rmi.Remote).	F

Regarding practice 6 (Java Message Service):

7. Since JMS is an example of non-persistent messaging, when a client is not connected it loses messages sent from other clients.	F
8. The first message sent by the CsdMessengerClient component to the "csd" queue indicates the name of the user.	T