B.E. INFORMATION TECHNOLOGY 4TH YEAR 2nd SEMESTER EXAMINATION - 2019

Subject: Distributed Systems: Applications

Time: 3 Hours **Full Marks: 100**

(Note: Answers of all parts/subparts of a question/group should be written together)

CO1 (30)	Q.1 Answer (a) or any *two* from (b), (c) and (d): a. Differentiate between Distributed Operating System (DOS), Network Operating System (NOS) and Distributed System (DS) with respect to the critical issues of them. 10 b. Suppose both client and server in a Distributed System (DS) are communicating by Remote Procedure Call (RPC). If the server is crashed before/after the execution of the request, sent by the client; what are the strategies that both the parties can take to overcome it. 10 c. What is the remote object reference? Define client stub and server stub. How these two entities work while two parties are communicating using Remote Method Invocation (RMI) in a Distributed System. 2+4+4 d. Differentiate between Persistent Asynchronous Communication and Persistent Synchronous Communication in Message Oriented Communication over Distributed System (DS). Describe the role of Message Broker in it. Is the architecture used by Message Queuing Model, used by Distributed System, different than the usual Queuing system, used in the Internet router? Give reasons. 4+3+3
CO2 (15)	Q.2 Answer any *three* from (a), (b), (c) and (d): a. What are the differences between Iterative and Recursive Name Server? 5 b. Explain how naming system is used by Remote Procedure Call (RPC) and Remote Method Invocation (RMI). 5 c. Discuss the absence of a syntactic distinction (such as use of a final '.') between absolute and relative names in DNS. 5 d. When might a DNS server provide multiple answers to a single name lookup, and why? 5
CO3 (30)	Q.3 Answer (a) and any *two* from (b), (c) and (d): a. Assuming that strict two-phase locking is in use, describe how the actions of the two-phase commit protocol relate to the concurrency control actions of each individual server. How does distributed deadlock detection fit in? 6+4 **[In the Q.3 (b)-(d), consider a server 'S' that manages the objects a_1, a_2, ..., a_n. The server provides two operations for its clients: *read* (*i*) returns the value of a_i; *write*(*i*, *Value*) assigns *Value* to a_i.]** b. Two transactions *T* and *U* at the server 'S' are defined as follows: *T*: $x = read(j)$; $y = read(i)$; $write(j, 44)$; $write(i, 33)$; *U*: $x = read(k)$; $write(i, 55)$; $y = read(j)$; $write(k, 66)$. Give three serially equivalent interleavings of the transactions *T* and *U*. 10 c. Consider a relaxation of two-phase locks in which read only transactions can release read locks early. Would a read only transaction have consistent retrievals? Would the objects become inconsistent? Illustrate your answer with the following transactions *T* and *U* at the 'S': *T*: x = read (i); y= read(j); *U*: write(i, 55);write(j, 66); where initial values of a_i and a_j are 10 and 20. 10 d. Consider a deadlock detection scheme for a single server. Describe precisely when edges are added to and removed from the wait-for-graph. Illustrate your answer with respect to the following transactions *T*, *U* and *V* at

	the server 'S'. | *T*	*U*	*V*
---	---	---
	write(i, 66);	
write(i, 55);		
		write(i, 77);
	commit		 When *U* releases its write lock on a_i, both *T* and *V* are waiting to obtain write locks on it. Does your scheme work correctly if *T* (first come) is granted the lock before *V*? If your answer is 'No', then modify your description. 10
CO4 (10)	Q.4 Answer either (a) or (b): a. Give some examples of faults in hardware and software that can/cannot be tolerated by the use of redundancy in a distributed system. To what extent does the use of redundancy in the appropriate cases make a system fault-tolerant? 5+5 b. What is tolerating omission failures? How Sliding window protocol can overcome it? 3+7
CO5 (15)	Q.5 Answer either (a) or (b): a. Sun NFS aims to support heterogeneous distributed systems by the provision of an operating system-independent file service. What are the key decisions that the implementer of an NFS server for an operating system other than UNIX would have to take? What constraints should an underlying filing system obey to be suitable for the implementation of NFS servers? What data must the NFS client module hold on behalf of each user-level process? 7+5+3 b. How many lookup calls are needed to resolve a 5-part pathname (for example, */usr/users/jim/code/xyz.c*) for a file that is stored on an NFS server? What is the reason for performing the translation step-by-step? Compare the update semantics of UNIX when accessing local files with those of NFS and AFS. Under what circumstances might clients become aware of the differences? 4+5+6

-: Course Objectives :-

CO1: Recollect Distributed Systems preliminaries, Distributed Communication Protocols and Fundamental Algorithms;
CO2: Illustrate and sketch different Name services and relate them in case study of Global Name Service;
CO3: Analyze, compare different Distributed Database handling issues like transaction, Concurrency, Consistency and Replication protocols;
CO4: Describe and analyze the concept of Fault Tolerance in Distributed Systems and compare different Failure Detection and Stabilization mechanisms;
CO5: Analyze, compare and distinguish different Distributed File Systems;