UNIT I

INTRODUCTION

PART A

S. No.	Question	CO	Blooms Taxanomy Level
1	**Define distributed system.** A distributed system is a collection of independent computers that appears to its users as a single coherent system. A distributed system is one in which components located at networked communicate and coordinate their actions only by passing message.	C309.1	**BTL-5**
2	**List the characteristics of distributed system?** • Programs are executed concurrently, support for resource sharing. • Openness • Concurrency • Scalability • Fault Tolerance (Reliability) • Transparency • Components can fail independently (isolation, crash)	C309.1	**BTL-1**
3	**Mention the examples of distributed system.** • The internet, intranet. • Department computing cluster • Corporate systems • Cloud systems (e.g. Google, Microsoft, etc.) • Mobile and ubiquitous computing	C309.1	**BTL-1**
4	**What is mobile and ubiquitous computing?** Mobile: computing devices are being carried around. Ubiquitous: little computing devices are all over the place.	C309.1	**BTL-1**
5	**Mention the challenges in distributed system.** 1. Heterogeneity 2. Openness 3. Security 4. Scalability	C309.1	**BTL-1**

	5. Failure handling 6. Concurrency 7. Transparency		
6	**What are the Advantages of Distributed Systems?** 1. Performance 2.Distribution 3.Reliability (fault tolerance) 4.Incremental growth 5.Sharing of data/resources 6.Communication	C309.1	**BTL-1**
7	**What are the Disadvantages of Distributed Systems?** MAY/JUNE 2016, **NOV/DEC 2016** 1.Difficulties of developing distributed software 2.Networking problems 3.Security problems **Software** Little software exists compared to PCs (for example) but the situation is improving with the cloud. **Networking** Still slow and can cause other problems (e.g., when disconnected) **Security** Data may be accessed by unauthorized users through network interfaces **Privacy** Data may be accessed securely but without the owner’s consent (significant issue in modern systems)	C309.1	**BTL-1**
8	**What are the Applications of Distributed system?** • Email • News • Multimedia information systems - video conferencing • Airline reservation system • BanKing system • File downloads (BitTorrent) • Messaging	C309.1	**BTL-1**
9	**Write the different trends in distributed systems?** • The emergence of pervasive networking technology; • The emergence of ubiquitous computing coupled with the desire to support user mobility in distributed systems;	C309.1	**BTL-1**

	• The increasing demand for multimedia services; • The view of distributed systems as a utility.		
10	**What are distributed multimedia systems?** A distributed multimedia system (DMS) is an integrated communication, computing, and information system that enables the processing, management, delivery, and presentation of synchronized multimedia information with quality-of-service guarantees.	C309.1	**BTL-1**
11	**What are different types of transparencies required in distributed systems? NOV/DEC 2017** • Access Transparency • Execution Transparency • Replication Transparency • Performance Transparency • Configuration Transparency	C309.1	**BTL-1**
12	**Write a short note on quality of service in distributed systems.** Quality of Service (a.K.a. QoS) refers to performance and other service expectations of a client or an application. • Performance • Reliability and availability • security Examples where this is important. • Voice over IP (VOIP) and telephony • Video (e.g. Netflix and friends)	C309.1	**BTL-2**
13	**Advantages of Distributed Systems vs. Centralized** • Economics • Speed • Geographic and Responsibility Distribution • Reliability • Extendibility	C309.1	**BTL-1**
14	**Write the Resource Sharing of Distributed system? NOV/DEC 2017** 1.Share hardware, 2.software, 3.data and information **Hardware Devices** Printers, disks, memory, sensors **Software Sharing** Compilers, libraries, toolkits, computational	C309.1	**BTL-1**

	Kernels **Data** Databases, files		
15	**What are the Design issues of Distributed system?** • Openness • Resource Sharing • Concurrency • Scalability • Fault-Tolerance • Transparency • High-Performance	C309.1	**BTL-1**
16	**Write the issues arising from Distributed Systems?** • Naming - How to uniquely identify resources. • Communication - How to exchange data and information reliably with good performance. • Software Structure - How to make software open, extensible, scalable, with high-performance. • Workload Allocation - Where to perform computations and various services. • Consistency Maintenance - How to Keep consistency at a reasonable cost.	C309.1	**BTL-1**
17	**What is Communication in Distributed Systems?** Communication is an essential part of distributed systems - e.g., clients and servers must communicate for request and response. Communication normally involved - transfer of data from sender to receiver - synchronization among processes.	C309.1	**BTL-1**
18	**What are types of Communication in Distributed Systems** • Client-Server • Group Multicast • Function Shipping • Performance of distributed systems depends critically on communication performance	C309.1	**BTL-1**
19	**Distributed System Software Structure** • It must be easy to add new services (flexibility, extensibility, openness requirements) • Kernel is normally restricted to o memory allocation o process creation and scheduling	C309.1	**BTL-1**

	o interposes communication peripheral device handling		
20	**What is concurrency in Distributed Systems?** Both services and applications provide resources that can be shared by clients in a distributed system. There is therefore a possibility that several clients will attempt to access a shared resource at the same time. For example, a data structure that records bids for an auction may be accessed very frequently when it gets close to the deadline time.	C309.1	**BTL-1**
21	**What is Web Search?** Web search has emerged as a major growth industry in the last decade, with recent figures indicating that the global number of searches has risen to over 10 billion per calendar month. The task of a web search engine is to index the entire contents of the World Wide Web, encompassing a wide range of information styles including web pages, multimedia sources and (scanned) books.	C309.1	**BTL-1**
22	**What is Transparency? APR/MAY 2017** Transparency is defined as the concealment from the user and the application programmer of the separation of components in a distributed system, so that the system is perceived as a whole rather than as a collection of independent components.	C309.1	**BTL-1**
23	**What are the failure handlings in Distributed Systems?** • Detecting failures • Masking failures • Tolerating failures • Recovery from failures	C309.1	**BTL-1**
24	**Define heterogeneity.** The Internet enables users to access services and run applications over a heterogeneous collection of computers and networks. Heterogeneity (that is, variety and difference) applies to all of the following: • Networks; • Computer hardware; • Operating systems; • Programming languages; • Implementations by different developers.	C309.1	**BTL-1**
25	**Define virtual machine.** The *virtual machine* approach provides a way of making code executable on a variety of host computers: the compiler for a particular language	C309.1	**BTL-1**

	generates code for a virtual machine instead of a particular hardware order code.		
26	**Define openness. APR/MAY 2017** The openness of distributed systems is determined primarily by the degree to which new resource-sharing services can be added and be made available for use by a variety of client programs.	C309.1	**BTL-1**
27	**What is security?** Security: Encryption can be used to provide adequate protection of shared Resources and to keep sensitive information secret when it is transmitted in messages over a network. Denial of service attacks are still a problem.	C309.1	**BTL-1**
28	**What is Scalability?** A distributed system is scalable if the cost of adding a user is a constant amount in terms of the resources that must be added. The algorithms used to access shared data should avoid performance bottlenecks and data should be structured hierarchically to get the best access times. Frequently accessed data can be replicated.	C309.1	**BTL-1**
29	**What is Failure handling?** Any process, computer or network may fail independently of the others. Therefore each component needs to be aware of the possible ways in which the components it depends on may fail and be designed to deal with each of those failures appropriately.	C309.1	**BTL-1**
30	**What is Concurrency?** The presence of multiple users in a distributed system is a source of concurrent requests to its resources. Each resource must be designed to be safe in a concurrent environment.	C309.1	**BTL-1**
31	**What is the role of middleware? MAY/JUNE 2016** The term middleware applies to a software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, operating systems and programming languages. In addition to solving the problems of heterogeneity, middleware provides a uniform computational model for use by the programmers of servers and distributed applications.	C309.1	**BTL-1**
32	**Define Transparency (APRIL/MAY 2017, NOV/DEC 2017)** Transparency is defined as the concealment from	C309.1	**BTL-1**

	the user and the application programmer of the separation of components in a distributed system, so that the system is perceived as a whole rather than as a collection of independent components		
33	**List any two resources of hardware and software, which can be shared in distributed systems with example. (NOV 2017)** Hardware – Printer, Disks, Fax machine, Router, Modem. Software – Application Programs, Shared Files, Shared Databases, Documents, Services.	C309.1	**BTL-1**
34	**State the objectives of resurce sharing model APRIL/MAY 2018** ▪ resources are made available ▪ resources can be used ▪ service provider and user interact with each other ▪ accessing remote resources ▪ sharing them in a controlled and efficient way	C309.1	**BTL-1**
35	**Write down the Principles of distributed systems APRIL/MAY 2018** The principles of distributed computing, emphasizing the fundamental issues underlying the design of distributed systems and networks: communication, coordination, fault-tolerance, locality, parallelism, self-organization, synchronization, uncertainty	C309.1	**BTL-1**
36	**What is Location aware computing? NOV/DEC 2018** Location-aware computing is a technology that uses the location of people and objects to derive contextual information with which to enhance the application behaviour. There are two ways to acquire information about user context: requiring the user to specify it or by monitoring users and computer activity. Sensor technology, such as RFID, could enable mobile devices to extract information from user position automatically.	C309.1	**BTL-1**
37	**State any three factors to be considered for variations in client server model. NOV/DEC 2018** • The use of mobile code and mobile agents. • User’s need for low-cost computers with limited h/w resources that are simple to manage. • The requirement to add and remove mobile devices in a convenient manner.	C309.1	**BTL-2**

	PART B		
1	Define distributed systems. What are the significant issues and challenges of the distributed systems? **NOV/DEC 2017, APRIL/MAY 2018 George Coulouris-*Pg-no- 16***	C309.1	BTL-6
2	Enlighten the examples of distributed systems. **MAY/JUNE 2016 George Coulouris-*Pg-no- 3***	C309.1	BTL-5
3	Write short notes on WWW. **MAY/JUNE 2016 , APRIL/MAY 2018 George Coulouris-*Pg-no- 26***	C309.1	BTL-1
4	Discuss the different trends in distributed systems. **MAY/JUNE 2016, NOV/DEC 2016, NOV/DEC 2017, APRIL MAY 2018, George Coulouris-*Pg-no- 8***	C309.1	BTL-5
5	What is resource sharing? Explain. **NOV/DEC 2016 George Coulouris-*Pg-no- 14***	C309.1	BTL-5
6	What are the challenges in Distributed System? Explain. **NOV/DEC 2016 George Coulouris-*Pg-no- 16***	C309.1	BTL-5
7	What are the different types of failures and how are they dealt in distributed systems? **George Coulouris-*Pg-no- 21***	C309.1	BTL-5
8	What are different types of transparencies required in distributed systems? **George Coulouris-*Pg-no- 23***	C309.1	BTL-6
9	Write a short note on quality of service in distributed systems. **George Coulouris-*Pg-no- 25***	C309.1	BTL-5
10	What are different application domains of distributed systems? Explain. **George Coulouris-*Pg-no- 4***	C309.1	BTL-5
11	What is the need for distributed system? List the distributed systems challenges. **NOV/DEC 2018**	C309.1	BTL-5
12	Identify the five types of hardware resource and five types of data or software resource that can be shared efficiently. Give examples **NOV/DEC 2018**	C309.1	BTL-5
13.	i.Elaborate the design issues to be considered for spontaneous networking. ii. a user arrives at a railway station for first time, carrying a PDA that is capable of wireless networking. Suggest how the user could be provided with information about the local services and amenities at that station. Without entering the stations name or attributes. What are the technical challenges to be addressed? **NOV/DEC 2018**	C309.1	BTL-5

UNIT II

PART – A

S. No.	Question	CO	Blooms Taxanomy Level
1	**What is the architectural model?** An architectural model defines the way in which the components of system interact with one another and the way in which they are mapped onto an underlying network of computers**.**	C309.2	BTL-1
2	**What is the role of proxy server ? MAY/JUNE 2016** In computer networks, a **proxy server** is a **server** (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other **servers**.	C309.2	BTL-1
3	**Define inter process communication**. **MAY/JUNE 2016** **Interprocess communication** (IPC) is a set of programming interfaces that allow a programmer to coordinate activities among different program **process**es that can run concurrently in an operating system. This allows a program to handle many user requests at the same time.	C309.2	BTL-1
4	**What are different types of models for developing distributed systems?** o Interaction models o Failure models o Security models	C309.2	BTL-1
5	**What is Interaction Models, Failure models, Security models?** o Which consider the structure and sequencing of the communication between the elements of the system.	C309.2	BTL-1

	o Which consider the ways in which a system may fail to operate correctly. o Which consider how the system is protected against attempts to interfere with its correct operation or to steal its data.		
6	**What are the types of system model?** o Physical models o Architectural models o Fundamental models	C309.2	BTL-1
7	**What is Fundamental model?** Fundamental models that help to reveal key problems for the designers of distributed systems. Their purpose is to specify the design issues, difficulties and threats that must be resolved in order to develop distribute systems that fulfill their tasks correctly, reliably and securely.	C309.2	BTL-1
8	**What is Physical model?** A physical model is a representation of the underlying hardware elements of a distributed system that abstracts away from specific details of the computer and networking technologies employed.	C309.2	BTL-1

9	**What are the types of communication paradigm in DS?**
	o Interprocess communication;
	o Remote invocation;
	o Indirect communication.
		C309.2	BTL-1

10	**What are the communicating entities in DS?** o System-oriented entities o Problem oriented entities	C309.2	BTL-1
11	**What is mobile agent? MAY/JUNE 2016** A mobile agent is a running program (including both code and data) that travels from one computer to another in a network carrying out a task on someone's behalf, such as collecting information, and eventually returning with the results.	C309.2	BTL-1
12	**What is layering?** The concept of layering is a familiar one and is closely related to abstraction.In a layered approach, a complex system is partitioned into a number of layers, with a given layer making use of the services offered by the layer below.	C309.2	BTL-1
13	**What are Software and hardware service layers in distributed systems?** o Applications, services	C309.2	BTL-1

	o Middleware o Operating system o Computer and network hardware		
14	**What is the role of AJAX?** AJAX meets the need for fine-grained communication between a Javascript front-end program running in a web browser and a server-based back-end program holding data describing the state of the application	C309.2	BTL-1
15	**What is CDR?** CORBA CDR is the external data representation defined with CORBA 2.0. CDR can represent all of the data types that can be used as arguments and return values in remote invocations in CORBA. It consists of 15 primitive types that include short (16-bit), long (32-bit), unsigned short, unsigned long, float (32-bit), double(64-bit), char, Boolean (TRUE or FALSE), octet (8-bit) and any constructed types.	C309.2	BTL-1
16	**What are the major categories of middleware?** o Distributed objects o Distributed components o Publish-subscribe systems o Message queues o Web services o Peer-to-peer	C309.2	BTL-1
17	**Define clock drift rate.** The term clock drift rate refers to the rate at which a computer clock deviates from a perfect reference clock.	C309.2	BTL-1
18	**What are Omission failure and its types?** The faults classified as omission failures refer to cases when a process or communication channel fails to perform actions that it is supposed to do. Types are, 1. Process omission. 2. Communication omission.	C309.2	BTL-1
19	**What is reliable communication?** The term reliable communication is defined in terms of validity and integrity as follows: **Validity:** Any message in the outgoing message buffer is eventually delivered to the incoming message buffer. **Integrity:** The message received is identical to one sent, and no messages are delivered twice.	C309.2	BTL-1
20	**What are the system architecture types?** o Mini computer model o Workstation model o Workstation server model	C309.2	BTL-1

	o Processor pool model o Hybrid model		
21	**What are Difficulties and threats for distributed systems?** o Widely varying modes of use o Wide range of system environments o Internal problems o External threats	C309.2	BTL-1
22	**What is the difference between RMI and RPC? APR/MAY 2017** Remote Procedure Call or the RPC and the Remote Method Invocation or RMI are both message passing techniques in the Inter Process Communication (IPC). But there are two basic differences between the two methods: 1. RPC supports procedural programming. i.e. only remote procedures can be invoKed. Whereas RMI is object-based. As the name suggests, it is invoKed on remote objects. 2. In RPC, the parameters that are passed are ordinary data structures. Whereas in RMI, objects can be passed as parameters.	C309.2	BTL-5
23	**What is meant by inter process Communication?** Inter process communication is concerned with the communication between processes in a distributed system, both in its own right and as support for communication between distributed objects. The Java API for inter process communication in the internet provides both datagram and stream communication	C309.2	BTL-1
24	**What is meant by group communication?** Group communication is a multicast operation is more appropriate- this is an operation that sends a single message from one process to each of the members of a group of process, usually in such a way that the membership of the group is transparent to the sender.	C309.2	BTL-1
25	**What is the use of RMI registry?** The RMI registry is used to store a list of available services. A client uses the registry to make it's proxy object, and the Registry is responsible for giving appropriate information to the client so that it can hook up with the server that implements the service.	C309.2	BTL-1
26	**What is a Message queue?** Message queues offer a point-to-point service whereby producer processes can send messages to a specified queue and consumer processes can receive messages from the queue or be notified of the arrival of new messages in the queue. Queues therefore	C309.2	BTL-1

	offer an indirection between the producer and consumer processes.		
27	**What is a Tuple space?** Tuple spaces offer a further indirect communication service by supporting a model whereby processes can place arbitrary items of structured data, called tuples, in a persistent tuple space and other processes can either read or remove such tuples from the tuple space by specifying patterns of interest. Since the tuple space is persistent, readers and writers do not need to exist at the same time.	C309.2	BTL-1
28	**Difference between synchronous and asynchronous communication?** In **synchronous form of communication**, the sending and receiving processes synchronize at every message. In this case, both send and receive are blocking operations. Whenever a send is issued the sending process is blocked until the corresponding receive is issued. Whenever receive is issued, the process blocks until a message arrives. In **asynchronous form of communication**, the use of the send operation is non-blocking in that the sending process is allowed to proceed as soon as the message has been copied to a local buffer and the transmission of the message proceeds in parallel with the sending process. The receive operation can have blocking and non-blocking variants.	C309.2	BTL-5
29	**What is marshalling and unmarshalling**? Marshalling is the process of taking a collection of data items and assembling them into a form suitable for transmission in a message. Unmarshalling is the process of disassembling them on arrival to produce an equivalent collection of data items at the destination.	C309.2	BTL-1
30	**Define Datagram.** A datagram is, to quote the Internet's Request for Comments 1594, "a self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network." The term is used in several well-known communication protocols, including the User Datagram Protocol and AppleTalk.	C309.2	BTL-1
31	**What is remote object reference?** A remote object reference is an identifier for a remote object that is valid throughout a distributed system. A remote object reference is passed in the invocation message to specify which object is to be invoked.	C309.2	BTL-1

32	**What is Multicast operation?** A multicast operation is more appropriate – this is an operation that sends a single message from one process to each of the members of a group of processes, usually in such a way that the membership of the group is transparent to the sender.	C309.2	BTL-1
33	**What infrastructure provided by multicast message for distributed system?** o Fault tolerance based on replicated services o Discovering services in spontaneous networking o Better performance through replicated data o Propagation of event notifications	C309.2	BTL-1
34	**Define multicast router.** Internet multicasts make use of multicast routers, which forward single datagram's to routers on other networks, where they are again multicast to local members.	C309.2	BTL-1
35	**Define Request Reply Protocol.** Request-reply protocols provide lightweight and minimal support for client-server computing. Such protocols are often used in environments where overheads of communication must be minimized.	C309.2	BTL-1
36	**What is the use of UDP?** The Domain Name Service (DNS), which looks up DNS names in the Internet, is implemented over UDP. UDP datagram's are sometimes an attractive choice because they do not suffer from overheads associated with guaranteed message delivery.	C309.2	BTL-1
37	**What is meant by client server communication?** The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.	C309.2	BTL-1
38	**What are the features of TCP Protocol?** Sequencing Flow control Retransmission Buffering Check sum	C309.2	BTL-1
39	**What is the use of ports?** Port numbers are used for addressing messages to processes within a particular computer and are valid only within that computer. A port number is a 16-bit integer. Once an IP packet has been delivered to the destination host, the TCP- or UDP-layer software	C309.2	BTL-1

	dispatches it to a process via a specific port at that host.		
40	**What is Routing?** Routing is a function that is required in all networks except those LANs, such as Ethernets, that provide direct connections between all pairs of attached hosts. In large networks, adaptive routing is employed: the best route for communication between two points in the network is re-evaluated periodically, taking into account the current traffic in the network and any faults such as broken connections or routers.	C309.2	BTL-1
41	**Why restricted length packets are used?** PacKets of restricted length are used: • So that each computer in the network can allocate sufficient buffer storage to hold the largest possible incoming packet. • To avoid the undue delays that would occur in waiting for communication channels to become free if long messages were transmitted without subdivision.	C309.2	BTL-2
42	**List the different switching scheme?** Broadcast Circuit switching Packet switching Frame relay	C309.2	BTL-2
43	**List the types of packet delivery?** Datagram pacKet delivery Virtual circuit pacKet delivery	C309.2	BTL-1
44	**What is indirect communication?** Indirect communication is defined as communication between entities in adistributed system through an intermediary with no direct coupling between the sender and the receiver(s).	C309.2	BTL-1
45	**Define publish subscribe system.** Publish-subscribe systems, a family of approaches that all share the common characteristic of disseminating events to multiple recipients through an intermediary;	C309.2	BTL-1
46	**What are the Applications of publish-subscribe systems?** • Financial information systems; • Other areas with live feeds of real-time data (including RSS feeds); • Support for cooperative working, where a number of participants need to be informed of events of shared interest; • Support for ubiquitous computing, including the management of events emanating from the ubiquitous infrastructure	C309.2	BTL-1

	(for example, location events); • A broad set of monitoring applications, including network monitoring in the Internet.		
47	**Define distributed shared memory.** Distributed shared memory (DSM) is an abstraction used for sharing data between computers that do not share physical memory. Processes access DSM by reads and updates to what appears to be ordinary memory within their address space.	C309.2	BTL-1
48	**What are the three types of communication paradigm in distributed system? NOV 2017** The three types of communication paradigm are: Inter-process communication Remote invocation Indirect communication	C309.2	BTL-1
49	**Differentiate persistent and non-persistent connections? NOV 2017** HTTP can use both non-persistent connections and persistent connections. A non-persistent connection is the one that is closed after the server sends the requested object to the client. In other words, the connection is used exactly for one request and one response. Non-persistent connections are the default mode for HTTP/1.0 With persistent connections, the server leaves the TCP connection open after sending responses and hence the subsequent requests and responses between the same client and server can be sent. The server closes the connection only when it is not used for a certain configurable amount of time. Persistent connections are the default mode for HTTP/1.1.	C309.2	BTL-1
50	**List out the design issues for RPC? APRIL/MAY 2018** The style of programming promoted by RPC programming with interfaces; The call semantics associated with RPC; The key issue of transparency and how it relates to remote procedure calls.	C309.2	BTL-1
51	**Compare the communicating entities:object,components and web services. NOV/DEC 2018** **Object:** Distributed objects are objects (in the sense of object-oriented programming) that are distributed across different address spaces, either in different processes on the same computer, or even in multiple	C309.2	BTL-2

	computers connected via a network, but which work together by sharing data and invoking methods**.** **Components:** components are located on different networked computers, which then communicate and coordinate their actions by passing messages to one another. The components interact with one another in order to achieve a common goal. **web services:** Its a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web		
52	**Identify the two signifigant factors affecting interacting process in distributed systems.APRIL/MAY 2018** ■ Communication performance is often a limiting characteristic. ■ It is impossible to maintain a single global notion of time.	C309.2	BTL-2
53	**"Tiered architectures are complementary to layering"-comment. NOV/DEC 2018** Layering deals with vertical organization of services Tiered architectures are complementary to layering, which deals with horizontal organization of services.	C309.2	BTL-2
PART B			
1	Illustrate TCP and UDP communication with suitable example program. **MAY/JUNE 2016, NOV/DEC 2016 George Coulouris-*Pg-no- 122***	C309.2	BTL-5
2	Write down the steps in JavaRMI and explain it with suitable example. **MAY/JUNE 2016 George Coulouris-*Pg-no- 217***	C309.2	BTL6
3	What are different types of models for developing distributed systems? Explain each model in brief.**APRIL/MAY 2018 George Coulouris-*Pg-no- 39***	C309.2	BTL-5
4	What are fundamental models for distributed system? What is the purpose of such models **NOV/DEC 2017 George Coulouris-*Pg-no- 61***	C309.2	BTL-5
5	What are the characteristics of interprocess communications? Explain. **George Coulouris-*Pg-no- 147***	C309.2	BTL-2
6	Explain the Java API for UDP datagram and TCP stream communication. **George Coulouris-*Pg-no-150***	C309.2	BTL-5
7	What is CORBA's common Data Representation? Explain. **George Coulouris-*Pg-no-340***	C309.2	BTL4
8	What is object serialization? How does Java serialize objects? **George Coulouris-*Pg-no-337***	C309.2	BTL-2

9	What is network virtualization? What are overlay networks? What are advantages and disadvantages of overlay networks? **George Coulouris-*Pg-no-433***	C309.2	BTL6
10	Explain the request reply protocol along with its operations and message structure. **George Coulouris-*Pg-no-187***	C309.2	BTL-2
11	What ire Remote Procedure Call? Discuss the design issues for RPC **NOV/DEC 2017 George Coulouris-*Pg-no-195***	C309.2	BTL-6
12	What is remote method invocation? What are the commonalities and differences between RPC and RMI? And design issues for RMI? **George Coulouris-*Pg-no-204***	C309.2	BTL-5
13	What is Indirect Communication? Explain space and time coupling and uncoupling. **NOV/DEC 2016 George Coulouris-*Pg-no-230***	C309.2	BTL-5
14	What is group communication? What are the Key areas of applications of group communication? Explain the programming model for group communication.**APRIL/MAY 2018 George Coulouris-*Pg-no-232***	C309.2	BTL-5
15	What are publish subscribe systems? What are their applications? Explain with example. **George Coulouris-*Pg-no-242***	C309.2	BTL-5
16	Compare message passing and distributed shared memory approaches. **George Coulouris-*Pg-no-254***	C309.2	BTL-5
17	i.discuss on DSM with suitable illustrations. ii.Consider a simple server that carries out client requests without accessing other servers. Explain why it is generally not possible to set a limit on the time taken by a server to respond to a client request. What should the server do, to execute requests within a bounded time. **NOV/DEC 2018**	C309.2	BTL-5
18	i.What is publish subscribe systems? Explain its characteristics features with neat sketch. ii.Classify and tabulate the arbitrary failures with respect to class affects and comments. **NOV/DEC 2018**	C309.2	BTL-5

UNIT III

PART A			
S. No.	Question	CO	Blooms Taxanomy Level
1	**What is peer to peer system?** Peer-to-peer systems aim to support useful distributed services and applications using data and computing resources available in the personal computers and workstations that are present in the Internet and other networks in ever-increasing numbers.	C309.3	BTL-1
2	**What is goal of peer to peer system?** The goal of peer-to-peer systems is to enable the sharing of data and resources on a very large scale by eliminating any requirement for separately managed servers and their associated infrastructure.	C309.3	BTL-1
3	**What are the characteristics of peer to peer system? MAY/JUNE 2016** Their design ensures that each user contributes resources to the system. • Although they may differ in the resources that they contribute, all the nodes in a peer-to-peer system have the same functional capabilities and responsibilities. • Their correct operation does not depend on the existence of any centrally administered systems. • They can be designed to offer a limited degree of anonymity to the providers and users of resources.	C309.3	BTL-1
4	**What is napster file system? NOV/DEC 2018** The need for and the feasibility of a peer-to-peer solution were first demonstrated by the Napster file sharing system [Open Nap 2001] which provided a means for users to share files. Napster became very popular for music exchange. At its peak, several million users were registered and thousands were swapping music files simultaneously.	C309.3	BTL-1
5	**What is the need of peer to peer middleware system? NOV/DEC 2017**	C309.3	BTL-1

	Peer-to-peer middleware systems are designed specifically to meet the need for the automatic placement and subsequent location of the distributed objects managed by peer-to-peer systems and applications.		
6	**Write the Non-functional requirements of peer-to-peer middleware system?** o Global scalability o Load balancing o Optimization for local interactions between neighbouring peers o Accommodating to highly dynamic host availability	C309.3	BTL-1
7	**What is the role of routing overlays in peer to peer system? APR/MAY 2017** Peer-to-peer systems usually store multiple replicas of objects to ensure availability. In that case, the routing overlay maintains Knowledge of the location of all the available replicas and delivers requests to the nearest 'live' node (i.e. one that has not failed) that has a copy of the relevant object.	C309.3	BTL-1
8	**What are the tasks performed by routing overlay?** o Insertion of objects o Deletion of objects o Node addition and removal	C309.3	BTL-1
9	**What are the generations of peer to peer system?** Three generations of peer-to-peer system and application development can be identified. o The first generation was launched by the Napster music exchange service [OpenNap 2001]. o A second generation of file sharing applications offering greater scalability, anonymity and fault tolerance quickly followed including Freenet, Gnutella, Kazaa and BitTorrent o The third generation is	C309.3	BTL-1

<table>
<tr><td></td><td>characterized by the emergence of middleware layers for the application-independent management of distributed resources on a global scale.</td><td></td><td></td></tr>
<tr><td>10</td><td><b>What are the case studies used in overlay? NOV/DEC 2017</b>
○ <b>Pastry</b> is the message routing infrastructure deployed in several applications including PAST.
○ <b>Tapestry</b> is the basis for the Ocean Store storage system.</td><td>C309.3</td><td>BTL-1</td></tr>
<tr><td>11</td><td><b>Difference between Structured versus unstructured peer-to-peer systems.</b>
<table>
<tr><td></td><td>Structured peer-to-peer</td><td>Unstructured peer-to-peer</td></tr>
<tr><td>Advantages</td><td>Guaranteed to locate objects (assuming they exist) and can offer time and complexity bounds on this operation; relatively low message overhead.</td><td>Self-organizing and naturally resilient to node failure.</td></tr>
<tr><td>Disadvantages</td><td>Need to maintain often complex overlay structures, which can be difficult and costly to achieve, especially in highly dynamic environments</td><td>Probabilistic and hence cannot offer absolute guarantees on locating objects; prone to excessive messaging overhead which can affect scalability.</td></tr>
</table></td><td>C309.3</td><td>BTL-1</td></tr>
</table>

	.		
12	**What is web caching?** Web browsers generate HTTP GET requests for Internet objects like HTML pages, images, etc. These may be serviced from a browser cache on the client machine, from a proxy web cache (a service running on another computer in the same local network or on a nearby node in the Internet) or from the origin web server (the server whose domain name is included in the parameters of the GET request), depending on which contains a fresh copy of the object.	C309.3	BTL-1
13	**What is Squirrel?** The Squirrel web caching service performs the same functions using a small part of the resources of each client computer on a local network.	C309.3	BTL-1
14	**What is meant by distributed file system?** A distributed file system enables programs to store and access remote files exactly as they do local ones, allowing users to access files from any computer on a network.	C309.3	BTL-1
15	**What are the two basic file system used in distributed system?** • The Sun Network File System, NFS. • The Andrew File System, AFS.	C309.3	BTL-1
16	**What are the file system modules?** o Directory module: relates file names to file IDs. o File module: relates file IDs to particular files. o Access control module: checks	C309.3	BTL-1

	permission for operation requested. o File access module: reads or writes file data or attributes. o Block module: accesses and allocates disk blocks. o Device module: performs disk I/O and buffering.		
17	**Write the Characteristics of file systems?** File systems are responsible for the organization, storage, retrieval, naming, sharing and protection of files. They provide a programming interface that characterizes the file abstraction, freeing programmers from concern with the details of storage allocation and layout.	C309.3	BTL-1
18	**What are the 2 components of file system?** Files contain both data and attributes. The **data** consist of a sequence of data items (typically 8-bit bytes), accessible by operations to read and write any portion of the sequence. The **attributes** are held as a single record containing information such as the length of the file, timestamps, file type, owner's identity and access control lists.	C309.3	BTL-1
19	**Define metadata.** The term metadata is often used to refer to all of the extra information stored by a file system that is needed for the management of files. It includes file attributes, directories and all the other persistent information used by the file system.	C309.3	BTL-2
20	**What are the Distributed file system requirements?** Transparency and location transparency; performance, scalability, concurrency control, fault tolerance, file replication, hardware and operating system heterogeneity, consistency, security and efficiency.	C309.3	BTL-1
21	**What are the different forms of transparency are partially or wholly addressed by current file services?** o Access transparency o Location transparency o Mobility transparency o Performance transparency o Scaling transparency	C309.3	BTL-5

22	**Define File service architecture of AFS?** An architecture that offers a clear separation of the main concerns in providing access to files is obtained by structuring the file service as three components – a *flat file service*, a *directory service* and a *client module*.	C309.3	BTL-1
23	**What is Andrew File System?** Andrew is a distributed computing environment developed at Carnegie Mellon University (CMU) for use as a campus computing and information system. The design of the Andrew File System (henceforth abbreviated AFS) reflects an intention to support information sharing on a large scale by minimizing client-server communication.	C309.3	BTL-1
24	**Define File group identifier.** File group identifiers must be unique throughout a distributed system.a unique identifier can be generated by concatenating the 32-bit IP address of the host creating the new group with a 16-bit integer derived from the date, producing a unique 48-bit integer: *32 bits* *16 bits* *File group identifier*: IP address date	C309.3	BTL-5
25	**Write the two unusual design characteristics of AFS?** **Whole-file serving:** The entire contents of directories and files are transmitted to client computers by AFS servers (in AFS-3, files larger than 64 Kbytes are transferred in 64-Kbyte chunKs). **Whole-file caching**: Once a copy of a file or a chunK has been transferred to a client computer it is stored in a cache on the local disK. The cache contains several hundred of the files most recently used on that computer. The cache is permanent, surviving reboots of the client computer. Local copies of files are used to satisfy clients' open requests in preference to remote copies whenever possible.	C309.3	BTL-2
26	**What are the two software components that exist in AFS implementation?** AFS is implemented as two software components that exist as UNIX processes called Vice and Venus. Vice is the name given to the server software that runs as a user-level UNIX process in each server computer, and Venus is a user-level process that	C309.3	BTL-1

	runs in each client computer and corresponds to the client module in our abstract model.		
27	**Write the Key design issues for distributed file systems?** The Key design issues for distributed file systems are: • the effective use of client caching to achieve performance equal to or better than that of local file systems; • the maintenance of consistency between multiple cached client copies of files when they are updated; • recovery after client or server failure; • high throughput for reading and writing files of all sizes; Scalability	C309.3	BTL-1
28	**What is the need of Name service?** In a distributed system, names are used to refer to a wide variety of resources such as computers, services, remote objects and files, as well as to users. A name is needed to request a computer system to act upon a specific resource chosen out of many.	C309.3	BTL-1
29	**Define Identifier.** identifier is sometimes used to refer to names that are interpreted only by programs. Remote object references and NFS file handles are examples of identifiers. Identifiers are chosen for the efficiency with which they can be looked up and stored by software.	C309.3	BTL-1
30	**Define object address.** object’s address: a value that identifies the location of the object rather than the object itself.	C309.3	BTL-1
31	**What is Uniform Resource Identifiers?** Uniform Resource Identifiers • Uniform Resource Identifiers (URIs) came about from the need to identify resources on the Web, and other Internet resources such as electronic mailboxes.	C309.3	BTL-1
32	**What is Uniform Resource Locator?** Uniform Resource Locator (URL) is often used for URIs that provide location information and specify the method for accessing the resource.	C309.3	BTL-1
33	**What is Uniform Resource Names?** Uniform Resource Names: Uniform Resource Names (URNs) are URIs that are used as pure	C309.3	BTL-1

	resource names rather than locators.		
34	**What is caching?** In DNS and other name services, client name resolution software and servers maintain a cache of the results of previous name resolutions. When a client requests a name lookup, the name resolution software consults its cache. If it holds a recent result from a previous lookup for the name, it returns it to the client; otherwise, it sets about finding it from a server. That server, in turn, may return data cached from other servers.	C309.3	BTL-1
35	**Define Name space. APR/MAY 2017** A name space is the collection of all valid names recognized by a particular service. The service will attempt to look up a valid name, even though that name may prove not to correspond to any object.	C309.3	BTL-1
36	**What is Name resolution?** Name resolution is an iterative or recursive process whereby a name is repeatedly presented to naming contexts in order to look up the attributes to which it refers.	C309.3	BTL-1
37	**What is caching?** Caching • In DNS and other name services, client name resolution software and servers maintain a cache of the results of previous name resolutions. When a client requests a name lookup, the name resolution software consults its cache. If it holds a recent result from a previous lookup for the name, it returns it to the client; otherwise, it sets about finding it from a server. That server, in turn, may return data cached from other servers.	C309.3	BTL-1
38	**What is Global Name Service?** A Global Name Service (GNS) was designed and implemented by Lampson and colleagues at the DEC Systems Research Center [Lampson 1986] to provide facilities for resource location, mail addressing and authentication.	C309.3	BTL-1
39	**Discuss on LDAP. MAY /JUNE 2016** **LDAP** (Lightweight Directory Access Protocol) is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet.	C309.3	BTL-2
40	**Give tbe the characteristics of Peer-to-Peer systems? JUNE 2016, NOV 2017, APRIL/MAY 2018**	C309.3	BTL-2

	Its design ensures that each user contributes resources to the system. Although they may differ in the resources that they contribute, all the nodes in a peer-to-peer system have the same functional capabilities and responsibilities. Its correct operation does not depend on the existence of any centrally administered systems. They can be designed to offer a limited degree of anonymity to the providers and users of resources. A key issue for their efficient operation is the choice of an algorithm for the placement of data across many hosts and subsequent access to it in a manner that balances the workload and ensures availability without adding undue overheads.		
41	**What is naming and locating facility? (NOV 2017)** A name service stores information about a collection of textual names, in the form of bindings between the names and the attributes of the entities they denote, such as users, computers, services and objects. The collection is often subdivided into one or more naming contexts: individual subsets of the bindings that are managed as a unit. The major operation that a name service supports is to resolve a name that is, to look up attributes from a given name	C309.3	BTL-1
42	**Label the different forms of transparency in file services APRIL/MAY 2018** Access Transparency Location Transparency Concurrency Transparency Replication Transparency Fault Transparency Migration Transparency Performance Transparency Scaling Transparency	C309.3	BTL-1
43	**Demonstrate the use of name cache?** To increase the performance of name service To reduce overall system overhead	C309.3	BTL-1
44	**List out the goals of Global Naming Service(GNS)?** To handle an essentially arbitrary number of names and to serve an arbitrary number of administrative organizations A long lifetime	C309.3	BTL-1

	High availability Fault isolation Tolerance of mistrust		
45	**Develop how does AFS deal with the risk that callback messages may be lost?** Callbacks must be renewed before an open if a time T (typically on the order of a few minutes) has elapsed since thefile was cached without communication from the server. This is to deal with possible communication failures, which can result in the loss of callback messages.	C309.3	BTL-1
46	**What are venus and vice processes?** AFS is implemented as two software components that exist as UNIX processes called Vice and Venus. Vice is the name given to the server software that runs as a user-level UNIX process in each server computer, and Venus is a user-level process that runs in each client computer andcorresponds to the client module in abstract model. Venus manages the cache, removing the least recently used files when a new file is acquired from a server to make the required space if the partition is full. The Vice servers accept requests only in terms of fid s. Venus translates the pathnames supplied by clients into fid s using a step-by-step lookup to obtain the information from the file directories held in the Vice servers**.**	C309.3	BTL-1
47	**Explain the working of Andrew File system.** The design of the Andrew File System (henceforth abbreviated AFS) reflects an intention to support information sharing on a large scale by minimizing client-server communication. This is achieved by transferring whole files between server and client computers and caching them at clients until the server receives a more up-to-date version.	C309.3	BTL-1
48	**What are the requirements of distributed file system (or) Give the design issues of Distributed file system?** Transparency Concurrent file updates File replication Hardware and operating system heterogeneity Fault tolerance Consistency Security Efficiency	C309.3	BTL-1
49	**What is distributed file system?** A Distributed File System is a file system with distributed storage and users. DFS provides	C309.3	BTL-1

	transparency of location, access, and migration of files. DFS systems use cache replies for efficiency and fault tolerance.		
50	**What is high availability and high reliability?** High availability: DFS should continue to function even when partial failures occur due to the failure of one or more components, such as a communication link failure, a machine failure or a storage device crash. High reliability: In a good distributed file system, the probability of loss of stored data should be minimized as far as practicable.	C309.3	BTL-1
51	**Define Gnutella. NOV/DEC 2018** Gnutella (possibly by analogy with the GNU Project) is a large peer-to-peer network. It was the first decentralized peer-to-peer network of its kind, leading to other, later networks adopting the model.	C309.3	BTL-1
PART – B			
1	Explain about the file server architecture. **MAY/JUNE 2016, APRIL/MAY 2017 , APRIL/MAY 2018 George Coulouris-*Pg-no-530***	C309.3	BTL-5
2	With neat sketch explain Routing Overlays in detail. **MAY/JUNE 2016, NOV/DEC 2016, APRIL/MAY 2017, APRIL/MAY 2018 George Coulouris-*Pg-no-433***	C309.3	BTL-5
3	Write short notes on Napster and its legacy and Peer to Peer Middleware**. MAY/JUNE 2016, APRIL/MAY 2017 George Coulouris-*Pg-no-428***	C309.3	BTL-5
4	Explain about the Andrew file system. **NOV/DEC 2016 George Coulouris-*Pg-no-548***	C309.3	BTL-5
5	Discuss about the distributed file system. **George Coulouris-*Pg-no-522***	C309.3	BTL-5
6	Explain the following: Uniform resource identifiers, Uniform resource locators, Uniform resource names. **George Coulouris-*Pg-no-573***	C309.3	BTL-5
7	Explain the DNS name service and bind implementation of DNS. **George Coulouris-*Pg-no-569***	C309.3	BTL-6
8	What is name service? What are its requirements? What is name resolution? What are the different ways of resolving names? Explain. **NOV/DEC**	C309.3	BTL-5

	2016 George Coulouris-*Pg-no-569*		
9	Discuss on File Sharing Semantics. **NOV/DEC 2016 George Coulouris-*Pg-no-585***	C309.3	BTL-6
10	Briefly Discuss the architecture and server operation of NFS?	C309.3	BTL-6
11	Discuss in detail the design and implementation of name services and Domain Name services.	C309.3	BTL-6
12	Explain the requirements of Distributed file system?	C309.3	BTL-2
13	Explain in details the characteristics of file system	C309.3	BTL-2
14	Write a short note on(i)DNS(ii)Name Service	C309.3	BTL-1
15	Explain directory services	C309.3	BTL-2
16	what is pastry?explain the pastrys algorithm with pseudo code. **NOV/DEC 2018**	C309.3	BTL-6
17	i.explain file service architecture and Andrew file system with suitable sketch? ii.State the difference between overlay networking and ip routing **NOV/DEC 2018**	C309.3	BTL-6

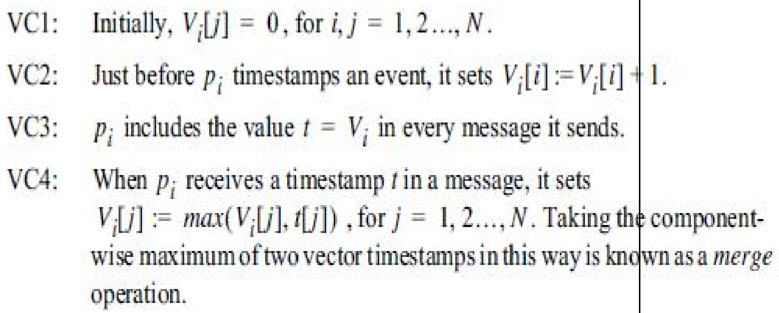
UNIT IV			
PART A			
S. No.	Question	CO	Blooms Taxanomy Level
1	**What is clock skew and clock drift? APRIL/MAY 2018** The instantaneous difference between the readings of any two clocks is called their skew. Clock drift means that they count time at different rates, and so diverge.	C309.4	BTL-2

2	**What is clocks drift rate?** A clock’s *driftrate* is the change in the offset (difference in reading) between the clock and a nominal perfect reference clock per unit of time measured by the reference clock.	C309.4	BTL-1
3	**What is meant by International Atomic Time ?** Computer clocks can be synchronized to external sources of highly accurate time. The most accurate physical clocks use atomic oscillators, whose drift rate is about one part in 1013. The output of these atomic clocks is used as the standard for elapsed real time, Known as *International Atomic Time.*	C309.4	BTL-1
4	**What is Coordinated Universal Time?** *Coordinated Universal Time* – abbreviated as UTC (from the French equivalent) – is an international standard for timeKeeping. It is based on atomic time, but a so-called ‘leap second’ is inserted – or, more rarely, deleted – occasionally to Keep it in step with astronomical time. UTC signals are synchronized and broadcast regularly from landbased radio stations and satellites covering many parts of the world.	C309.4	BTL-1
5	**What are the two modes of synchronization? Write their format?** The two modes are: **External synchronization:** In order to Know at what time of day events occur at the processes in our distributed system – for example, for accountancy purposes – it is necessary to synchronize the processes’ clocks, *Ci* , with an authoritative, external source of time. This is *externalsynchronization* For a synchronization bound D>0, and for a source S of UTC time,$\lvert S(t) – Ci(t)\rvert<T$, for i=1,2,…N and for all real times t in I. **Internal synchronization:** If the clocks *Ci* are synchronized with one another to a Known degree of accuracy, then we can measure the interval between two events occurring at different computers by appealing to their local clocks, even though they are not necessarily synchronized to an external source of time. This is *internal synchronization*.	C309.4	BTL-1

	For a synchronization bound $D>0,\lvert Ci(t)-Cj(t)\rvert<D$, for i,j=1,2,…N. and for all real times t in I.		
6	**Explain Faultry and Crash Failure.** A clock that does not Keep to whatever correctness conditions apply is defined to be *faulty*. A clock's *crash failure* is said to occur when the clock stops ticKing altogether;any other clock failure is an *arbitrary failure*. A historical example of an arbitrary failure is that of a clock with the 'Y2K bug', which broKe the monotonicity condition by registering the date after 31 December 1999 as 1 January 1900 instead of 2000; another example is a clock whose batteries are very low and whose drift rate suddenly becomes very large	C309.4	BTL-2
7	**How the clock synchronization done in Cristian's method?** A single time server might fail, so they suggest the use of a group of synchronized servers . It does not deal with faulty servers	C309.4	BTL-1
8	**Explain the BerKley algorithm.** The BerKeley algorithm eliminates readings from faulty clocks. Such clocks could have a significant adverse effect if an ordinary average was taken so instead the master takes a *fault-tolerant average*. That is, a subset is chosen of clocks that do not differ from one another by more than a specified amount, and the average is taken of readings from only these clocks.	C309.4	BTL-1
9	**What are the issues resolved by BerKley's algorithm?** The collection of computers whose clocks are to be synchronized are categorized as masters and slaves. The averaging of the clock values cancels out the individuals clocks tendencies to run fast or slow. This overcomes the uncertainty due to message transmission time introduced in the Synchronized clock values returned by the master.	C309.4	BTL-1
10	**What is network partition?** The network partition can be used to separate a group of replica managers into two or more sub groups. The members of same subgroup communicate with one another but members of	C309.4	BTL-1

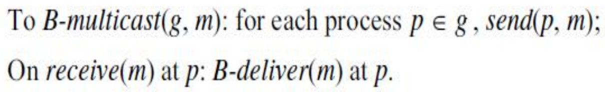
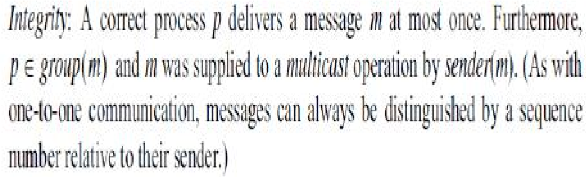
	different subgroup cannot communicate with one another		
11	**Write the features of Network Time Protocol** To provide a service enabling clients across the Internet to be synchronizedaccurately to UTC To provide a reliable service that can survive lengthy losses of connectivity: To enable clients to resynchronize sufficiently frequently to offset the rates of driftfound in most computers: To provide protection against interference with the time service, whether maliciousor accidental:	C309.4	BTL-1
12	**What are the different modes of synchronization of NTP servers.** NTP servers synchronize with one another in one of three modes: multicast, procedure-call and symmetric mode. ***Multicast mode*** is intended for use on a high-speed LAN. One or more servers periodically multicasts the time to the servers running in other computers connected by the LAN, which set their clocks assuming a small delay.This mode can achieve only relatively low accuracies, but ones that nonetheless are considered sufficient for many purposes. ***Procedure-call mode*** is similar to the operation of Cristian’s algorithm,. In this mode, one server accepts requests from other computers, which it processes by replying with its timestamp (current clock reading). This mode is suitable where higher accuracies are required than can be achieved with multicast, or where multicast is not supported in hardware. For example, file servers on the same or a neighbouring LAN that need to Keep accurate timing information for file accesses could contact a local server in procedure-call mode. ***Symmetric mode*** is intended for use by the servers that supply time information in LANs and by the higher levels (lower strata) of the synchronization subnet, where the highest accuracies are to be achieved. A pair of servers operating in symmetric mode exchange messages bearing timing information. Timing data are retained as part of an association between the	C309.4	BTL-1

	servers that is maintained in order to improve the accuracy of their synchronization over time.		
13	**Difference between reliable and unreliable failure detector?** Reliable failure detector is one that is always accurate n detecting a process failure. It answers processes queries with either a response of unsuspected-which, as before can only be a hint-or failed. U Reliable failure detector may produce one of two values when given the identity of a process: Unsuspected or suspected. Both of these results are hints, which may or may not accurately reflect whether the process has actually failed.	C309.4	BTL-1
14	**Define election algorithm? Mention the different algorithm?** An algorithm for choosing a unique process to play a particular role is called an election algorithm. Ex: In a variant of central server algorithm for mutual exclusion, the server is choosen from among the process . The different algorithms are • Ring based election algorithm • Bully algorithm.	C309.4	BTL-1
15	**Explain Logical time and logical clocks. MAY/JUNE 2016** **Logical time** Lamport proposed a model of *logical time* that can be used to provide an ordering among the events at processes running in different computers in a distributed system.Logical time allows the order in which the messages are presented to be inferred without recourse to clocks. **Logical clocks •** Lamport invented a simple mechanism by which the happenedbefore ordering can be captured numerically, called a *logical clock*. A Lamport logical clock is a monotonically increasing software counter, whose value need bear no particular relationship to any physical clock. Each process *pi* Keeps its own logical clock, *Li* , which it uses to apply so-called *Lamport timestamps* to events. We denote the timestamp of event *e* at *pi* by *Li(e)* , and by *L(e)* we denote the timestamp of event *e* at whatever process it occurred at.	C309.4	BTL-2

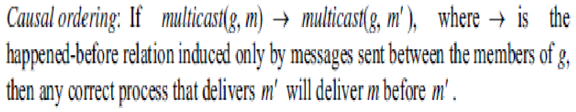
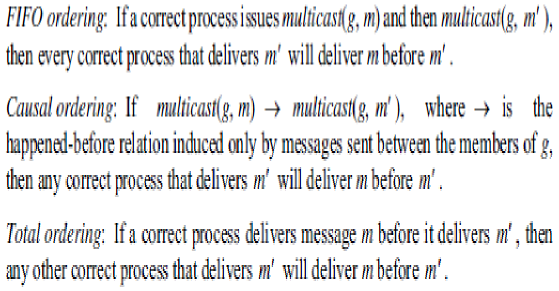
16	**What is vector clock? Explain.** **Vector clocks •** Mattern and Fidge developed vector clocks to overcome the shortcoming of Lamport’s clocks: the fact that from *L*(*e*)<*L*(*e’*)we cannot conclude that *e* ->*e*’ A vector clock for a system of *N* processes is an array of *N*integers. Each process Keeps its own vector clock, *Vi* , which it uses to timestamp local events. Like Lamport timestamps, processes piggybacK vector timestamps on the messages they send to one another, and there are simple rules for updating the clocks:  VC1: Initially, $V_i[j] = 0$, for $i, j = 1, 2..., N$. VC2: Just before p_i timestamps an event, it sets $V_i[i] := V_i[i] + 1$. VC3: p_i includes the value $t = V_i$ in every message it sends. VC4: When p_i receives a timestamp t in a message, it sets $V_i[j] := max(V_i[j], t[j])$, for $j = 1, 2..., N$. Taking the componentwise maximum of two vector timestamps in this way is known as a *merge* operation. TaKing the componentwise maximum of two vector timestamps in this way is Known as a *merge* operation.	C309.4	BTL-2
17	**Explain global states and consistent cuts with example.** Global state of a distributed system consists of –Local stateof each process: messages sent and messages received –State of each channel:messages sent but not received	C309.4	BTL-2
18	**Explain the ‘snapshot’ algorithm of Chandy and Lamport.** The ‘snapshot’ algorithm of Chandy and Lamport describe a ‘snapshot’ algorithm for determining global states of distributed systems, which we now present. The goal of the algorithm is to record a set of process and channel states (a ‘snapshot’) for a set of processes *pi*(*i* = 1,2,.*N*) such that, even though the combination of recorded states may never have occurred at the same time, the recorded global state is consistent.	C309.4	BTL-1
19	**Write about the rules of Chandy and Lamport’s ‘snapshot’ algorithm** The algorithm is defined through two rules, the	C309.4	BTL-1

	marker receiving rule and the *marker sending rule* The marker sending rule obligates processes to send a marker after they have recorded their state, but before they send any other messages. The marker receiving rule obligates a process that has not recorded its state to do so. In that case, this is the first marker that it has received. It notes which messages subsequently arrive on the other incoming channels. When a process that has already saved its state receives a marker (on another channel), it records the state of that channelas the set of messages it has received on it since it saved its state		
20	**What is a failure detector? What are its types? Explain.** A *failure detector* is a service that processes queries about whether a particular process has failed. It is often implemented by an object local to each process (on thesame computer) that runs a failure-detection algorithm in conjunction with its counterparts at other processes. The object local to each process is called a *local failure detector*. We outline how to implement failure detectors shortly, but first we concentrate on some of the properties of failure detectors. *Unreliable failure detectors*. An unreliable failure detector may produce one of two values when given the identity of a process: *Unsuspected* or *Suspected*. Both of these results are hints, which may or may not accurately reflect whether the process has actually failed. A *reliable failure detector* is one that is always accurate in detecting a process's failure. It answers processes' queries with either a response of *Unsuspected* – which, as before, can only be a hint – or *Failed*. A result of *Failed* means that the detector has determined that the process has crashed	C309.4	BTL-1
21	**What is mutual exclusion? How is the performance of algorithms of mutual exclusions evaluated?** Distributed processes often need to coordinate their activities. If a collection of processes share a resource or collection of resources, then often mutual exclusion is required to prevent	C309.4	BTL-1

	interference and ensure consistency when accessing the resources.		
22	**Explain the central server algorithm for mutual exclusion.** **The central server algorithm •** The simplest way to achieve mutual exclusion is to employ a server that grants permission to enter the critical section.	C309.4	BTL-2
23	**Explain the ring based algorithm for mutual exclusion** **A ring-based algorithm •** One of the simplest ways to arrange mutual exclusion between the *N* processes without requiring an additional process is to arrange them in a logical ring.	C309.4	BTL-2
24	**Explain an algorithm using multicast and logical clocks for mutual exclusion.** **An algorithm using multicast and logical clocks •** An algorithm to implement mutual exclusion between *N* peer processes that is based upon multicast. The basic idea is that processes that require entry to a critical section multicast a request message, and can enter it only when all the other processes have.	C309.4	BTL-1
25	**Explain MaeKawa's voting algorithm.** **MaeKawa's voting algorithm •** MaeKawa observed that in order for a process to enter a critical section, it is not necessary for all of its peers to grant it access. Processes need only obtain permission to enter from *subsets* of their peers, as long as the subsets used by any two processes overlap	C309.4	BTL-1
26	**Explain ring based election algorithm.** **A ring-based election algorithm •** The algorithm of Chang and Roberts is suitable for a collection of processes arranged in a logical ring. Each process *pi* has a communication channel to the next process in the ring $P(i+1) mod N$, and all messages are sent clockwise around the ring. The goal of this algorithm is to elect a single process called the *coordinator*, which is the	C309.4	BTL-2

	process with the largest identifier.		
27	**Explain the bully algorithm NOV/DEC 2017** This algorithm allows process to crash during an election. Although it assumes that message delivery between process is reliable. It assume sthat the system is synchronous – it uses timeouts to detect a process failure.	C309.4	BTL-2
28	**What is the use of Multicast operation?** It is generally useful to allow processes to be members of several groups simultaneously – for example, to enable processes to receive information from several sources by joining several groups	C309.4	BTL-1
29	**Explain the basic multicast algorithm.** . We allow processes to belong to several groups, and each message is destined for some particular group.. We call the primitive *B-multicast* and its corresponding basic delivery primitive *B-deliver* A straightforward way to implement *B-multicast* is to use a reliable one-to-one*send* operation, as follows:  To *B-multicast*(*g*, *m*): for each process $p \in g$, *send*(*p*, *m*); On *receive*(*m*) at *p*: *B-deliver*(*m*) at *p*.	C309.4	BTL-1
30	**What is reliable multicast algorithm. ?** We define a *reliable multicast* with corresponding operations *R-multicast* and *R-deliver*. A reliable multicast is one that satisfies the following properties:  *Integrity*: A correct process *p* delivers a message *m* at most once. Furthermore, $p \in group(m)$ and *m* was supplied to a *multicast* operation by *sender*(*m*). (As with one-to-one communication, messages can always be distinguished by a sequence number relative to their sender.) *Validity*: If a correct process multicasts message *m*, then it will eventually deliver *m*. *Agreement*: If a correct process delivers message *m*, then all other correct processes in *group*(*m*) will eventually deliver *m*.	C309.4	BTL-1

31	**How is reliable multicast implemented over basic multicast and IP multicast?** To *R-multicast* a message, a process *Bmulticast*s the message to the processes in the destination group (including itself). When the message is *B-deliver*ed, the recipient in turn *B-multicast*s the message to the group (if it is not the original sender), and then *R-deliver*s the message. Since a message may arrive more than once, duplicates of the message are detected and not delivered.This algorithm clearly satisfies the validity property, since a correct process will eventually *B-deliver* the message to itself. By the integrity property of the underlying communication channels used in *B-multicast*, the algorithm also satisfies the integrity property.	C309.4	BTL-2
32	**Explain the different types of ordering of multicasts messages in non-overlapping groups.** **Reliable multicast over IP multicast •** An alternative realization of *R-multicast* is to use a combination of IP multicast, piggybacked acknowledgements (that is, acknowledgements attached to other messages) and negative acknowledgements. This *R-multicast* protocol is based on the observation that IP multicast communication is often successful.	C309.4	BTL-2
33	**Explain FIFO ordering of multicast messages.** **FIFO ordering:** if a correct process issues multicast(g,m) and then multicast(g,m’) then every correct process that delivers m’ will deliver m before m’.	C309.4	BTL-1
34	**Explain total ordering of multicast messages.** Total ordering: if a correct process delivers message m before it delivers m’, then any other correct process that delivers m’ will deliver m before m’.	C309.4	BTL-1
35	**How is causal ordering implemented for ordering of multicast messages? Explain.**	C309.4	BTL-1

	 *Causal ordering*: If $multicast(g, m) \rightarrow multicast(g, m')$, where $\rightarrow$ is the happened-before relation induced only by messages sent between the members of g, then any correct process that delivers m' will deliver m before m'.		
36	**Explain the different types of ordering of multicasts messages in overlapping groups.**  *FIFO ordering*: If a correct process issues $multicast(g, m)$ and then $multicast(g, m')$, then every correct process that delivers m' will deliver m before m'. *Causal ordering*: If $multicast(g, m) \rightarrow multicast(g, m')$, where $\rightarrow$ is the happened-before relation induced only by messages sent between the members of g, then any correct process that delivers m' will deliver m before m'. *Total ordering*: If a correct process delivers message m before it delivers m', then any other correct process that delivers m' will deliver m before m'.	C309.4	BTL-2
37	**What is a Transaction?** A transaction defines a sequence of server operations that is guaranteed by the server to be atomic in the presence of multiple clients and server crashes. **What is flat transaction?** In a flat transaction, a client makes requests to more than one server. Transaction *T* is a flat transaction that invokes operations on objects in servers *X*, *Y* and *Z*. A flat client transaction completes each of its requests before going on to the next one. Therefore, each transaction accesses servers' objects sequentially.When servers use locking, a transaction can only be waiting for one object at a time.	C309.4	BTL-1
38	**What is meant by nested transactions? APR/MAY 2017** Nested transactions are structured from sets of other transactions. They are particularly useful in distributed systems because they allow additional concurrency.	C309.4	BTL-1
39	**Write the three methods of transactions .** • Locks are used to order transactions that access the same objects according to the order of arrival of their operations at the objects. • Optimistic concurrency control allows	C309.4	BTL-1

	transactions to proceed until they are ready to commit, whereupon a check is made to see whether they have performed conflicting operations on objects. • Timestamp ordering uses timestamps to order transactions that access the same		
40	**What is atomic transaction?** All or nothing: A transaction either completes successfully, in which case the effects of all of its operations are recorded in the objects, or (if it fails or is deliberately aborted) has no effect at all.	C309.4	BTL-1
41	**What is atomic commit protocol** A transaction comes to an end when the client requests that it be committed or aborted. A simple way to complete the transaction in an atomic manner is for the coordinator to communicate the commit or abort request to all of the participants in the transaction and to Keep on repeating the request until all of them have acknowledged that they have carried it out. This is an example of a *one phase atomic commit protocol*	C309.4	BTL-1
42	**What are the operations in the Coordinator interface** *openTransaction() - trans;* Starts a new transaction and delivers a unique TID *trans*. This identifier will be used in the other operations in the transaction. *closeTransaction(trans)- (commit, abort);* Ends a transaction: a *commit* return value indicates that the transaction has committed; an *abort* return value indicates that it has aborted. *abortTransaction(trans);*Aborts the transaction.	C309.4	BTL-1
43	**Explain the two-phase commit protocol.** The two-phase commit protocol ***Phase 1 (voting phase):*** 1. The coordinator sends a *canCommit?* request to each of the participants in the transaction. 2. When a participant receives a *canCommit?* request it replies with its vote (*Yes* or *No*) to the coordinator. Before voting *Yes*, it prepares to commit by saving objects in permanent storage. If the vote is *No*, the participant aborts immediately. ***Phase 2 (completion according to outcome of vote):*** 3. The coordinator collects the votes (including its own).	C309.4	BTL-2

	(a) If there are no failures and all the votes are *Yes*, the coordinator decides to commit the transaction and sends a *doCommit* request to each of the participants. (b)Otherwise, the coordinator decides to abort the transaction and sends *doAbort* requests to all participants that voted *Yes*. 4. Participants that voted *Yes* are waiting for a *doCommit* or *doAbort* request from the coordinator. When a participant receives one of these messages it acts accordingly and, in the case of commit, makes a *haveCommitted* call as confirmation to the coordinator.		
44	**Operations for two-phase commit protocol** *canCommit?(trans) - Yes / No* Call from coordinator to participant to ask whether it can commit a transaction.Participant replies with its vote. *doCommit(trans)* Call from coordinator to participant to tell participant to commit its part of a transaction. *doAbort(trans)* Call from coordinator to participant to tell participant to abort its part of a transaction. *haveCommitted(trans, participant)* Call from participant to coordinator to confirm that it has committed the transaction. *getDecision(trans)- Yes / No* Call from participant to coordinator to ask for the decision on a transaction when it has voted *Yes* but has still had no reply after some delay. Used to recover from server crash or delayed messages.	C309.4	BTL-2
45	**Explain two-phase commit protocol for nested transactions**. A two-phase commit protocol is needed for nested transactions – it allows servers of provisionally committed transactions that have crashed to abort them when they recover	C309.4	BTL-2
46	**Explain hierarchic two-phase commit protocols.** **Hierarchic two-phase commit protocol •** In this approach, the two-phase commit protocol becomes a multi-level nested protocol. The coordinator of the top-level transaction communicates with the coordinators of the subtransactions for which it is the immediate	C309.4	BTL-2

	parent.		
47	**What is the use of Locking?** In a distributed transaction, the locks on an object are held locally (in the same server).The local lock manager can decide whether to grant a lock or make the requesting transaction wait. However, it cannot release any locks until it Knows that the transaction has been committed or aborted at all the servers involved in the transaction. When locking is used for concurrency control, the objects remain locked and are unavailable.	C309.4	BTL-1
48	**What are the different ways to control concurrency in distributed transactions? Explain with examples.** Locking In a distributed transaction, the locks on an object are held locally (in the same server). The local lock manager can decide whether to grant a lock or make the requesting transaction wait. However, it cannot release any locks until it Knows that the transaction has been committed or aborted at all the servers involved in the transaction. When locking is used for concurrency control, the objects remain locked and are unavailablefor other transactions during the atomic commit protocol, although an aborted transaction releases its locks after phase 1 of the protocol.	C309.4	BTL-2
49	**What is Timestamp ordering concurrency control?** In a single server transaction, the coordinator issues a unique timestamp to each transaction when it starts. Serial equivalence is enforced by committing the versions of objects in the order of the timestamps of transactions that accessed them. In distributed transactions, we require that each coordinator issue globally unique timestamps	C309.4	BTL-1
50	**What is Optimistic concurrency control?** Recall that with optimistic concurrency control, each transaction is validated before it is allowed to commit. Transaction numbers are assigned at the start of validation and transactions are serialized according to the order of the transaction numbers. A distributed transaction is validated by a collection of independent servers, each of whichvalidates transactions that access its own objects. This validation takes place during the first phase of the two-phase commit protocol.	C309.4	BTL-1

51	**What is distributed deadlock? Explain with example.** With deadlock detection schemes, a transaction is aborted only when it is involved in a deadlock. Most deadlock detection schemes operate by finding cycles in the transaction wait-for graph. In a distributedsystem involving multiple servers being accessed by multiple transactions, a global wait-for graph can in theory be constructed from the local ones. There can be a cycle in the global wait-for graph that is not in any single local one – that is, there can be a *distributed deadlock*	C309.4	BTL-1
52	**Explain phantom deadlocks.** A deadlock that is 'detected' but is not really a deadlock is called phantom deadlock. In distributed deadlock detection, information about wait-for relationships between transactions is transmitted from on server to another. If there is a deadlock, the necessary information will eventually be collected in one place and a cycle will be detected. Ja this procedure will take some time, there is a chance that one of the transactions that Holds a lock will meanwhile have released it, in which case the deadlock will no longer exist.	C309.4	BTL-2
53	**Explain edge chasing deadlock detection technique in distributed systems.** A distributed approach to deadlock detection uses a technique called *edge chasing* or *path pushing*. In this approach, the global wait-for graph is not constructed, but each of the servers involved has Knowledge about some of its edges. The servers attempt to find cycles by forwarding messages called *probes*, which follow the edges of the graph throughout the distributed system. A probe message consists of transaction wait-for relationships representing a path in the global wait-for graph.	C309.4	BTL-2
54	**How are transactions recovered in distributed systems?** Atomic property of transactions can be described in two aspects:	C309.4	BTL-2

	• Durability: objects are saved in permanent storage and will be available indefinitely thereafter. Acknowledgement of a client’s commit request implies that all the effects of the transaction have been recorded in permanent storage as well as in the server’s volatile object. • Failure atomicity: the effects of transactions are atomic even when the server crashes. • Both can be realized by recovery manager.		
55	**What is Replication?** Replication is a Key to providing high availability and fault tolerance in distributed systems. High availability is of increasing interest with the tendency towards mobile computing and consequently disconnected operation. Fault tolerance is an abiding concern for services provided in safety-critical and other important systems.	C309.4	BTL-1
56	**Write about motivations for replication . APRIL/MAY 2018** *Performance enhancement*: The caching of data at clients and servers is by now familiar as a means of performance enhancement. *Increased availability*: Users require services to be highly available. That is, the proportion of time for which a service is accessible with reasonable response times should be close to 100%. *Fault tolerance*: Highly available data is not necessarily strictly correct data. It may be out of date, for example; or two users on opposite sides of a network partition may make updates that conflict and need to be resolved.	C309.4	BTL-1
57	**What is Replication Transparency?** A common requirement when data are replicated is for *replication transparency*. That is, clients should not normally have to be aware that multiple *physical* copies of data exist. As far as clients are concerned, data are organized as	C309.4	BTL-1

	individual *logical* objects and they identify only one item in each case when they request an operation to be performed.		
58	**What are replica managers?** Components that contain the replicas on a given computer and perform operations upon them directly	C309.4	BTL-1
59	**When replica manager requires t be a state machine.** We sometimes require each replica manager to be a *state machine*. Such a replica manager applies operations to its replicas atomically (indivisibly), so that its execution is equivalent to performing operations in some strict sequence. Moreover, the state of its replicas is a deterministic function of their initial states and the sequence of operations that it applies to them.	C309.4	BTL-2
60	**What is Linearizability ?** There are various correctness criteria for replicated objects. The most strictly correct systems are *linearizable*, and this property is called *linearizability*.	C309.4	BTL-1
61	**What is sequential consistency ?** Aweaker correctness condition is *sequential consistency*, which captures an essential requirement concerning the order in which requests are processed without appealing to real time. The definition Keeps the first criterion from the definition for linearizability but modifies the second.	C309.4	BTL-1
62	**When replicated shared objects are said to be sequentially consistent?** A replicated shared object service is said to be sequentially consistent if *for any execution* there is some interleaving of the series of operations issued by all the clients that satisfies the following two criteria: • The interleaved sequence of operations meets the specification of a (single) correct copy of the objects. • The order of operations in the interleaving is consistent with the program order in which each individual client executed them.	C309.4	BTL-2
63	**What is Passive (primary-backup) replication?** In the *passive* or *primary-backup* model of	C309.4	BTL-1

	replication for fault tolerance,there is at any one time a single primary replica manager and one or more secondary replica managers – ‘backups’ or ‘slaves’. In the pure form of the model, front ends communicate only with the primary replica manager to obtain the service. The primary replica manager executes the operations and sends copies of the updated data to the backups. If the primary fails, one of the backups is promoted to act as the primary.		
64	**Define Active replication** In the *active* model of replication for fault tolerance, the replica managers are state machines that play equivalent roles and are organized as a group.Front ends multicast their requests to the group of replica managers and all the replica managers process the request independently but identically and reply. If any replica manager crashes, this need have no impact upon the performance of the service, since the remaining replica managers continue to respond in the normal way.	C309.4	BTL-1
65	**What is the use of gossip architecture ?** *Gossip architecture is* a frameworK for implementing highly available services by replicating data close to the points where groups of clients need it. The name reflects the fact that the replica managers exchange ‘gossip’ messages periodically in order to convey the updates they have each received from clients	C309.4	BTL-1
66	**What are the different operations of gossip service?** A gossip service provides two basic types of operation: *queries* are read-only operations and *updates* modify but do not read the state (the latter is a more restricted definition than the one we have been using).	C309.4	BTL-2
67	**What are Gossip messages ?** Replica managers send gossip messages containing information concerning one or more updates so that other replica managers can bring their state upto-date. A replica manager uses the entries in its timestamp table to estimate which updates any other replica manager has not yet received.	C309.4	BTL-1

68	**What is The Coda file system?** The Coda file system is a descendent of AFS that aims to address several requirements that AFS does not meet – particularly the requirement to provide high availability despite disconnected operation	C309.4	BTL-1
69	**What is CVV?** A CVV is a vector timestamp with one element for each server in the relevant VSG.Coda's replication strategy is optimistic – it allowsmodification of files to proceed when the network is partitioned or during disconnected operation. It relies on the attachment to each version of a file of a *Coda version vector*(CVV).	C309.4	BTL-1
70	**What are the techniques used to synchronize clocks? (NOV 2017)** Clocks can be synchronized both externally and internally. The various techniques are: □ Synchronization in a synchronous system. □ Christian's method for synchronizing clocks □ The Berkeley Algorithm □ The Network Time Protocol	C309.4	BTL-1
71	**Write down the types of messages in bully algorithm. (NOV 2017)** There are three types of message in this algorithm: □ An election message is sent to announce an election; □ An answer message is sent in response to an election message and □ A coordinator message is sent to announce the identity of the elected process	C309.4	BTL-1
72	**List the methods to ensure serializability.NOV/DEC 2018** Serializability is the classical concurrency scheme. It ensures that a schedule for executing concurrent transactions is equivalent to one that executes the transactions serially in some order. It assumes that all accesses to the database are done using read and write operations. A schedule is called ``correct" if we can find a serial schedule that is ``equivalent" to it.	C309.4	BTL-2
73	**State the issues in Clocks. NOV/DEC 2018** The Importance of Accurate Time on Computer Networks. The synchronization of time on computers and networks is often vitally important. Without it, the time on individual computers will slowly drift away from each other at varying degrees until potentially each has a	C309.4	BTL-2

	significantly different time.		
	PART B		
1	Explain the terms: Clocks, Clock Skew, Clock Drift, Co-ordinated Universal Time. **George Coulouris-*Pg-no-597***	C309.4	BTL-2
2	What are the different ways of synchronizing physical clocks? Explain **George Coulouris-*Pg-no-599***	C309.4	BTL-5
3	Explain Christian’s method for synchronizing clocks. **George Coulouris-*Pg-no-601***	C309.4	BTL-6
4	Explain the BerKley algorithm **George Coulouris-*Pg-no-603***	C309.4	BTL-6
5	What is Network Time Protocol? What are its aims and features? **George Coulouris-*Pg-no-603***	C309.4	BTL-6
6	Explain Logical time and logical clocks. **George Coulouris-*Pg-no-607***	C309.4	BTL-5
7	Explain global states and consistent cuts with example. **George Coulouris-*Pg-no-610***	C309.4	BTL-6
8	Explain the ‘snapshot’ algorithm of Chandy and Lamport. **APRIL/MAY 2017, APRIL/MAY 2018 George Coulouris-*Pg-no-***	C309.4	BTL-6
9	Explain the bully algorithm. **George Coulouris-*Pg-no-610***	C309.4	BTL-6
10	What is mutual exclusion? How is the performance of algorithms of mutual exclusions evaluated? **NOV/DEC 2016, APRIL/MAY 2017 , NOV/DEC 2017 George Coulouris-*Pg-no-633***	C309.4	BTL-6
11	Explain the central server algorithm for mutual exclusion. **George Coulouris-*Pg-no-635***	C309.4	BTL-5
12	Explain the ring based algorithm for mutual exclusion **George Coulouris-*Pg-no-636***	C309.4	BTL-5
13	Explain an algorithm using multicast and logical clocks for mutual exclusion. **George Coulouris-*Pg-no-637***	C309.4	BTL-5
14	Explain MaeKawa’s voting algorithm. **MAY/JUNE 2016 George Coulouris-*Pg-no-638***	C309.4	BTL-5
15	What is an election algorithm? Explain **NOV/DEC 2016**	C309.4	BTL-5

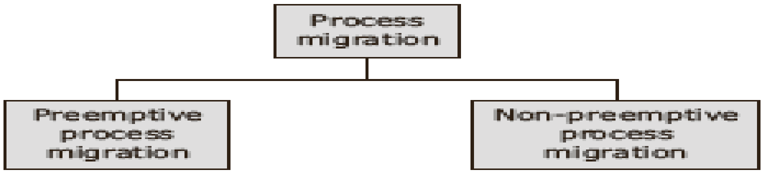
	ring based election algorithm. **George Coulouris-*Pg-no-641***		
16	Discuss transactions with suitable example. And Briefly explain Nested Transactions. **MAY/JUNE 2016 , APRIL/MAY 2018George Coulouris-*Pg-no-679&690***	C309.4	BTL-5
17	What is concurrency control? Explain in detail. **George Coulouris-*Pg-no-740***	C309.4	BTL-5
18	Write short notes on locks with suitable example. **George Coulouris-*Pg-no-692***	C309.4	BTL-5
19	What is a deadlock? How deadlock can be recovered? Explain distributed dead locks. **George Coulouris-*Pg-no-700***	C309.4	BTL-5
20	Explain optimistic concurrency control in detail? **APRIL/MAY 2018 George Coulouris-*Pg-no-707***	C309.4	BTL-5
21	Explain Two Phase commit protocol. **APRIL/MAY 2017 George Coulouris-*Pg-no-732***	C309.4	BTL-5
22	Discuss about timestamp ordering? **George Coulouris-*Pg-no-711***	C309.4	BTL-5
23	Explain the basic model for the management of replicated data . **George Coulouris-*Pg-no-768***	C309.4	BTL-5
24	Explain passive replication model for fault tolerance. **NOV/DEC 2016**	C309.4	BTL-5
25	Discuss active replication model **George Coulouris-*Pg-no-778***	C309.4	BTL-5
26	Explain architectures for replicated transaction. **George Coulouris-*Pg-no-783***	C309.4	BTL-6
27	What is atomic commit protocol? Explain the different protocols? **NOV/DEC 2016 George Coulouris-*Pg-no-731***	C309.4	BTL-5
28	How to control the concurrency in distributed system? **MAY/JUNE 2016 George Coulouris-*Pg-no-740***	C309.4	BTL-5
29	Explain CODA in detail. **APRIL/MAY 2017 ,**Elucidate coda architecture with respect to file systems, communication coda and processes in	C309.4	BTL-5

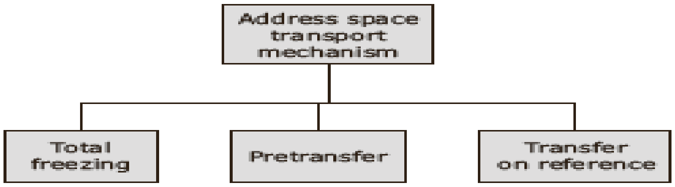
	coda with necessary diagrams. **NOV/DEC 2018**		
30	i.state the problems in cristians algorithm.explain how Berkeley algorithm overcomes the problems of cristians algorithm with neat sketch. ii.Describe the central server algorithm with neat sketch. State its performance measures. **NOV/DEC 2018**	C309.4	BTL-5

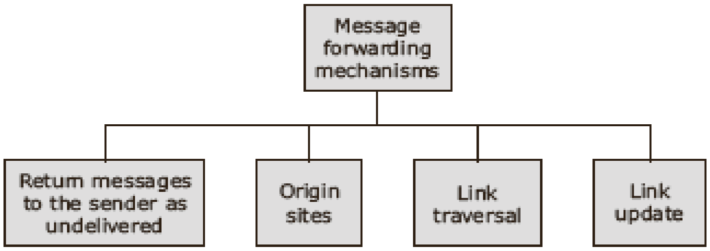
UNIT V

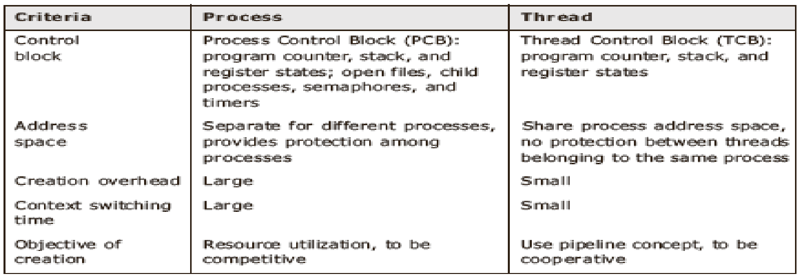
PART A

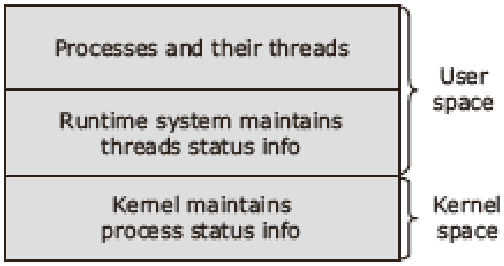
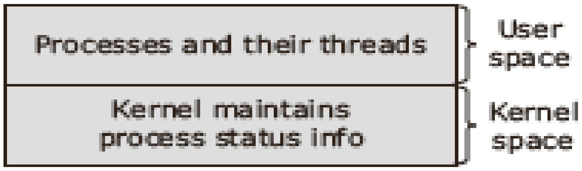
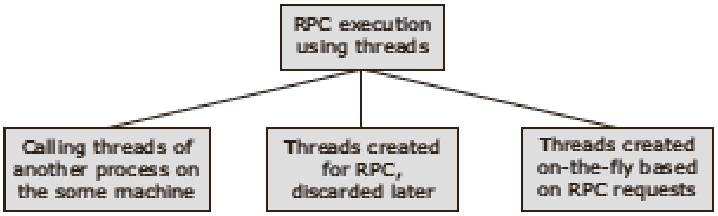
S. No.	Question	CO	Blooms Taxanomy Level
1	**What is the main goal of process management in DS?** ➨ Main goal of process management in DS is to make best possible use of existing resources by providing mechanism and polices for sharing them among processors ➨ **This is achieved by providing :** • **Process allocation** : decide which process should assign to which process in any instance of time for better utilization of resources . • **Process migration** : move process to new node for better utilization of resources • **Thread facilities** : provide mechanism for parallelism for better utilization of processor capabilities.	C309.5	BTL-1
2	**Explain the main functions of distributed process management .** ➨ **Process allocation :**decide which process should assign to which process in any instance of time	C309.5	BTL-1

	➨ **Process Migration:** Change of location and execution of a process from current processor to the destination processor for better utilization of resources and balancing load in distributed system.		
3	**How Process Migration is classified?** **Process Migration classified into :** ➨ **Non-preemptive : process is migrate before start execution in source node .** ➨ **Preemptive : process is migrate during its execution .**  Process migration Preemptive process migration Non-preemptive process migration	C309.5	BTL-1
4	**What are the desirable features of a good process migration mechanism?** ➨ Transparency: ▪ Object access level ▪ System call and IPC communication level • Location independent ➨ Minimal interference: ▪ To the progress of process and system ▪ Freezing time : time period for which the execution of the process is stopped for transferring its info to destination node ➨ Minimal residual dependencies: ▪ Migrated process should not depend on its previous node ➨ Efficiency: ▪ Issues • Time required to migrate process • Cost of locating the object • Cost of supporting remote execution once the process is migrated ➨ Robustness : ▪ Failure of any other node should not affect the accessibility or execution of the process ➨ Ability to communicate between co processes of the job: ▪ Communication directly possible irrespective of location	C309.5	BTL-2

5	**Explain Process Migration.What are the different steps involved in process migration?** **Process migration** is a specialized form of **process** management whereby **processes** are moved from one computing environment to another. **Steps involved in process migration** ➢ Freezing the process on its source and restarting it on its destination ➢ Transferring the process's address space (program code – data –stacK program) from its source to destination ➢ Forwarding messages meant for the migrant process ➢ Handling communication between cooperating process	C309.5	BTL-2
6	**Write about the Address space transport mechanisms.** ➨ **Total Freezing** ▪ Process's execution is stopped while transferring the address space ▪ Disadvantage that process may be suspended for a long time  Address space transport mechanism Total freezing Pretransfer Transfer on reference ➨ **Total Freezing** ▪ Process's execution is stopped while transferring the address space ▪ Disadvantage that process may be suspended for a long time ➨ **Pretransfering or Precopying** ▪ Address space transferred while process is running on the source node ➨ **Pretransfering or Precopying** ▪ Address space transferred while process is running on the source node ▪ After decision for migration is made process continues to execute on source node until address space is has been transferred ➨ **Transfer on Reference** ▪ Process executes on destination ▪ Address space is left behind in source node	C309.5	BTL-1

	▪ Desired blocks are copied from remote locations as and when required ▪ Failure of source node results in failure of process		
7	**What is Messages Forwarding?**  Message forwarding mechanisms Return messages to the sender as undelivered Origin sites Link traversal Link update ➨ Track and forward messages which have arrived on source node after process migration	C309.5	BTL-1
8	**Explain Return message to sender as undeliverable message mechanism.** ➨ Message type 1 and 2are returned to sender or dropped ➨ Sender retries after locating the new node (using locate operation) ➨ Type 3 message directly sent to new node	C309.5	BTL-1
9	**What is Origin site mechanism?** ➨ All messages are sent to origin site ➨ Origin site forwards the messages ➨ If origin site fails forwarding mechanism fails ➨ Continuous load on the origin site	C309.5	BTL-1
10	**What is LinK traversal mechanism?** ➨ Message queue is generated at origin ➨ Message Forwarded to destination node ➨ After process is migrated link is left on the previous node ➨ Process address has two parts process id, last Known location of destination node	C309.5	BTL-1
11	**What are the advantages of process migration ?** ➢ Reduce average response time of heavily loaded nodes ➢ Speed up of individual jobs ➢ Better utilization of resources	C309.5	BTL-1

<table>
<tr><td></td><td>➢ Improve reliability of critical processes</td><td></td><td></td></tr>
<tr><td>12</td><td><b>Comparison of processes and threads</b>
<table><tr><th>Criteria</th><th>Process</th><th>Thread</th></tr><tr><td>Control block</td><td>Process Control Block (PCB): program counter, stack, and register states; open files, child processes, semaphores, and timers</td><td>Thread Control Block (TCB): program counter, stack, and register states</td></tr><tr><td>Address space</td><td>Separate for different processes, provides protection among processes</td><td>Share process address space, no protection between threads belonging to the same process</td></tr><tr><td>Creation overhead</td><td>Large</td><td>Small</td></tr><tr><td>Context switching time</td><td>Large</td><td>Small</td></tr><tr><td>Objective of creation</td><td>Resource utilization, to be competitive</td><td>Use pipeline concept, to be cooperative</td></tr></table></td><td>C309.5</td><td>BTL-1</td></tr>
<tr><td>13</td><td><b>What are Thread models ?</b>
• Dispatcher worker model
• Team model
• Pipeline model</td><td>C309.5</td><td>BTL-1</td></tr>
<tr><td>14</td><td><b>What is Dispatcher worker model?</b>
Typical example of this model is server process such as file server that :
Accept request from client .
• ChecK for access permission.
• Accordingly services the request.</td><td>C309.5</td><td>BTL-1</td></tr>
<tr><td>15</td><td><b>What is Team Model.?</b>
All threads are treated equal , such that each one handle request on its own .
➨ In case threads are capable of performing specific distinct function , a queue can be maintained .
➨ When thread change state from running to idle , it take new request from the job queue and stars execution .
•</td><td>C309.5</td><td>BTL-1</td></tr>
<tr><td>16</td><td><b>What are the design issues in threads?</b>
➨ Thread semantics
• Thread creation, termination
➨ Thread synchronization
➨ Thread scheduling
•</td><td>C309.5</td><td>BTL-1</td></tr>
<tr><td>17</td><td><b>What are the different thread scheduling policies/algorithms</b>
• Priority assignment facility
• Choice of dynamic variation of quantum size</td><td>C309.5</td><td>BTL-1</td></tr>
</table>

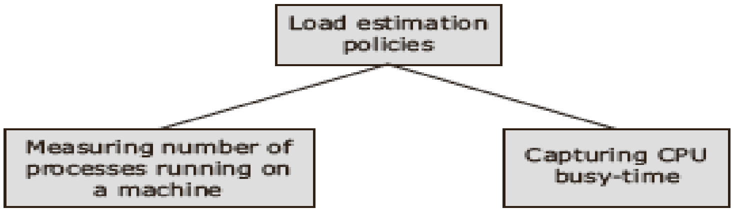
	• Handoff scheduling scheme • Affinity scheduling scheme • Signals used for providing interrupts and exceptions		
18	**How to implement thread package?** ➨ **Thread package can be implemented either in** ▪ user space ▪ Kernel space .  Processes and their threads Runtime system maintains threads status info Kernel maintains process status info User space Kernel space  Processes and their threads Kernel maintains process status info User space Kernel space	C309.5	BTL-2
19	**What are the different ways of Remote execution of thread?** **There are two different ways of remote execution of thread :** ➨ RPC : distributed systems commonly use RPC (remote procedure call). ➨ RMI (remote method invocation) and Java threads  RPC execution using threads Calling threads of another process on the some machine Threads created for RPC, discarded later Threads created on-the-fly based on RPC requests	C309.5	BTL-2
20	**What are the Component of faults?** ➨ **The computer is made up of :** ▪ Hardware components :processors , memory , I/O devices . ▪ Software components : OS , application programs m database . Distributed system made up of independent computers connected over the network , can fail due to fault in any of hardware components or software components.	C309.5	BTL-1
21	**What are the different types of components faults?**	C309.5	BTL-1

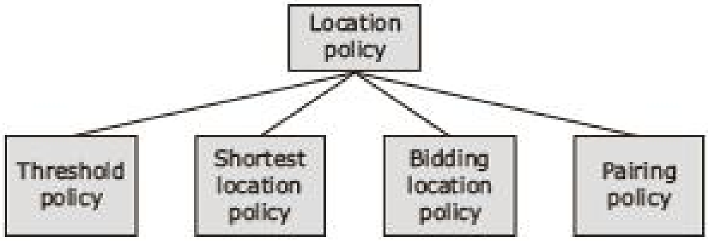
	- Transient faults - Intermittent faults - Permanent faults		
22	**What are the desirable features of Global Scheduling Algorithm?APRIL/MAY2018** No A Priori Knowledge about the Processes Ability to make dynamic scheduling decisions Flexible Stable Scalable Unaffected by system failures	C309.5	BTL-2
23	**What is Task Assignment Approach?** Each process is divided into multiple tasks. These tasks are scheduled to suitable processor to improve performance. This is not a widely used approach because:It requires characteristics of all the processes to be Known in advance.This approach does not take into consideration the dynamically changing state of the system.In this approach, a process is considered to be composed of multiple tasks and the goal is to find an optimal assignment policy for the tasks of an individual process.	C309.5	BTL-1
24	**What are the typical assumptions for the task assignment approach?** - A process has already been split into pieces called tasks. - The amount of computation required by each task and the speed of each processor are Known. - The cost of processing each task on every node of the system is Known. - The Interprocess Communication (IPC) costs between every pair of tasks is Known. - Other constraints, such as resource requirements of the tasks and the available resources at each node, precedence relationships among the tasks, and so on, are also Known	C309.5	BTL-1
25	**Write the Task Assignment Approach Algorithms.** ➨ Graph Theoretic Deterministic Algorithm. ➨ Centralized Heuristic Algorithm. ➨ Hierarchical Algorithm.	C309.5	BTL-1

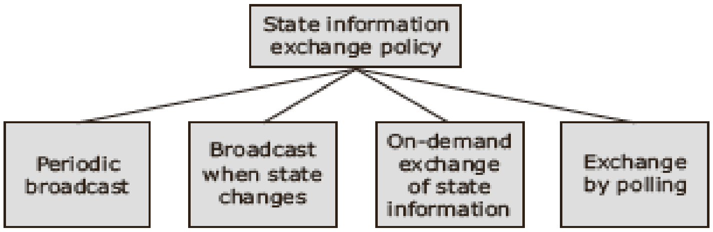
26	**What is Centralized Heuristic Algorithm?** Also called Top down algorithm.Doesn’t require advance information .Coordinator maintains the usage table with one entry for every user (processor) and this is initially zero.Usage table entries can either be zero, positive, or negative. Zero value indicates a neutral state, a positive value implies that the machine is user of system resources, and a negative value means that the machine needs resources	C309.5	BTL-1
27	**What is the Goal of Load Balancing Algorithms? MAY/JUNE 2016** ➨ The goal of the load balancing algorithms is to maintain the load to each processing element such that all the processing elements become neither overloaded nor idle that means each processing element ideally has equal load at any moment of time during execution to obtain the maximum performance (minimum execution time) of the system.	C309.5	BTL-1
28	**What is Load balancing?** Load balancing is the way of distributing load units (jobs or tasks) across a set of processors which are connected to a network which may be distributed across the globe. ➨ The excess load or remaining unexecuted load from a processor is migrated to other processors which have load below the threshold load. ➨ Threshold load is such an amount of load to a processor that any load may come further to that processor. ➨ By load balancing strategy it is possible to make every processor equally busy and to finish the worKs approximately at the same time.	C309.5	BTL-1
29	**What are the rules of Load Balancing Operation ?** A Load Balancing Operation is defined by three rules • Location Rule • This rule determines the partners of balancing operation i.e. The processors to involve in the balancing operation. • Distribution Rule • This rule determines how to redistribute workload among the processors in the balancing domain.	C309.5	BTL-1

	• Selection Rule • The selection rule works either in preemptive or in nonpreemptive fashion.		
30	**What are the Benefits of Load Balancing?** • Load balancing improves the performance of each node and hence the overall system performance • Higher throughput, reliability • Extensibility and incremental growth • Low cost but high gain • Load balancing reduces the job idle time • Response time becomes shorter • Maximum utilization of resources • Small jobs do not suffer from long starvation	C309.5	BTL-1
31	**What is Static Load Balancing?** • In static algorithm the processes are assigned to the processors at the compile time according to the performance of the nodes. • Once the processes are assigned, no change or reassignment is possible at the run time. Number of jobs in each node is fixed in static load balancing algorithm. Static algorithms do not collect any information about the nodes	C309.5	BTL-1
32	**What is Sub-optimal load balancing algorithm?** ➨ Sub-optimal load balancing algorithm will be mandatory for some applications when optimal solution is not found. ➨ In case the load is unpredictable or variable from minute to minute or hour to hour.	C309.5	BTL-2
33	**What is Dynamic Load Balancing?** ➨ In dynamic load balancing algorithm assignment of jobs is done at the runtime. ➨ In DLB jobs are reassigned at the runtime depending upon the situation that is the load will be transferred from heavily loaded nodes to the lightly loaded nodes. ➨ In dynamic load balancing no decision is taken until the process gets execution. ➨ This strategy collects the information about the system state and about the job information. ➨ As more information is collected by an algorithm in a short time, potentially the algorithm can make better decision.	C309.5	BTL-1

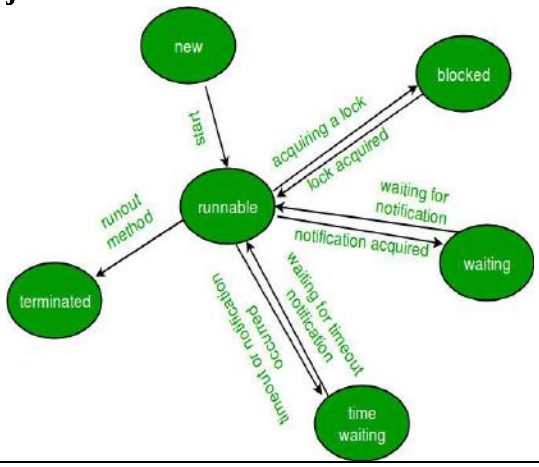
34	**What is Centralized Vs Distributed scheduling algorithm?** ➨ A **centralized** dynamic scheduling algorithm means that the scheduling decision is carried out at one single node called the centralized node. This approach is efficient since all information is available at a single node. ➨ DrawbacK for **centralized** approach is it leads to a bottleneck as number of requests increase . ➨ In a **distributed** algorithm the task of processor assignment is physically distributed among various nodes . In **distributed algorithm** scheduling decisions are made on individual nodes	C309.5	BTL-2
35	**What is Cooperative Vs non-cooperative Algorithm?** ➨ **Cooperative Algorithm** distributed entities cooperate with each other to make scheduling decisions ➨ **Non-Cooperative algorithm** the individual entities make independent scheduling decisions and hence they involve minor overheads ➨ Cooperative algorithms are more complex than non-cooperative ones ➨ Non-cooperative algorithms may not be stable	C309.5	BTL-2
36	**What are the issues in designing in load balancing algorithms? MAY/JUNE 2016** ➨ Deciding policies for: • Load estimation: determines how to estimate the worKload of a node in a distributed system. • Process transfer: decides whether the process can be executed locally or there is a need for remote execution. • Static information exchange: determines how the system load information can be exchanged among the nodes. • Location Policy: determines the selection of a destination node during process migration • Priority assignment: determines the priority of execution of a set of local and remote processes on a particular node	C309.5	BTL-2
37	**What are the policies for Load estimation?**	C309.5	BTL-1

	 Load estimation policies Measuring number of processes running on a machine Capturing CPU busy-time ➨ One way to estimate the load is to calculate the number of processes running on the machine ➨ The other technique is to capture the CPU busy-time. ➨ A machine with 75% CPU utilization is more heavily loaded than a machine with 40% CPU utilization. ➨ CPU utilization can be measured be measured by allowing a timer to interrupt the machine to periodically observe the CPU state and find the fraction of idle time**.** **Write the policies for Process transfer.** ➨ Load balancing strategy involves transferring some processes from heavily loaded nodes to lightly nodes ➨ So there is a need to decide a policy which indicates whether a node is heavily or lightly loaded, called threshold policy. ➨ This threshold is a limiting value which decides whether a new process, ready for execution or transferred to a lightly loaded node ➨ Threshold policy • Static • Dynamic		
38	**What are the policies of threshold?** • **Static:** each node has a predefined threshold value depending on its processing capability, which does not vary with load. The advantage of this method is that there is no need for any exchange of state information to decide the threshold value • **Dynamic:** the threshold value is calculated as an average worKload of all nodes. This policy gives a realistic value but involves the overhead of state information exchange among nodes to determine the threshold value.	C309.5	BTL-1
39	**What is Single-level threshold?** A node accepts a new process as long as its load is below the threshold, else it rejects the process as well as the requests for remote execution.	C309.5	BTL-1

	➨ The use of this policy may lead to useless process transfer, leading to instability in decisions. This may happen in a situation where a node accepts a new or a remote process when the load is below threshold , but its load may become larger than threshold just when the remote process arrives for execution.		
40	**What is Two-level threshold policy?** this policy is preferred to avoid instability it has two threshold levels: high and low marKs. The load states of a node can be divided into three regions: overloaded, normal and under-loaded	C309.5	BTL-1
41	**What are Location policies?** ➨ Once the threshold transfer policy decides to transfer a process from a node, the next step is to use a location policy to select the destination node where the process can be executed.  Location policy Threshold policy Shortest location policy Bidding location policy Pairing policy	C309.5	BTL-1
42	**What is Threshold policy ?** In the threshold policy, the destination node is selected at random and a checK is made to verify whether the remote process transfer would load that node, If not, the process transfer is carried out; else another node is selected at random and probed.	C309.5	BTL-1
43	**What is Shortest Location policy?** ➨ Nodes are chosen at random and each of these nodes is polled to checK for load. ➨ The node with the lowest load value is selected as the destination node. ➨ Once selected the destination node has to execute the process irrespective of its state at the time the process arrives. ➨ In case non of the polled nodes can accept the process it will be executed at the source node itself	C309.5	BTL-1

44	**What is Bidding Location Policy?** ➨ This policy transforms the system into a marKet scenario with buyers and sellers of services. ➨ Each node is assigned two roles, namely the manager and the contractor. ➨ The manager is an under loaded node having a process which needs a location and the contractor is a node which can accept remote processes. ➨ The manager broadcasts a request for a bid message to all nodes and the contractors send their bids to the manager node ➨ The bid contains information about processing power, and memory size	C309.5	BTL-1
45	**Define Pairing Policy.** ➨ The policies discussed earlier focus on load balancing across the systems, while the pairing policy focuses on load balancing between a pair of nodes ➨ Two nodes which have a large difference of load balancing between a pair of nodes are paired together temporarily ➨ The load balancing is carried out between the nodes belonging to the same pair by migrating processes from the heavily loaded node to the lightly loaded node ➨ Several pairs of nodes can exist in the system simultaneously	C309.5	BTL-1
46	**What is State information exchange policy?** ➨ Dynamic policies require frequent exchange of state information among the nodes of a system ➨ Decision based on state information  State information exchange policy Periodic broadcast Broadcast when state changes On-demand exchange of state information Exchange by polling	C309.5	BTL-1

47	**Write about Location policy for load sharing approach.** ➨ **Sender Initiated algorithm** uses sender of the process to decide where to send the process ➨ **Receiver initiated location policy** ➨ In this policy lightly loaded nodes search for heavily loaded nodes from which processes can be accepted for execution .	C309.5	BTL-1
48	**What are the advantages of process migration? MAY/JUNE 2016** Various advantages of process migration are: • Reduces average response time of processes. • Speeds up individual jobs. • Gains higher throughput. • Effective utilization of resources and reduces network traffic.	C309.5	BTL-1
49	**What is process migration? APRIL/MAY 2017** Process migration is the relocation of a process from its current location (source node) to another node(destination node).The process can be either a non-preemptive or preemptive process. Selection of the process to be migrated, selection of the destination node and the actual transfer of the selected process are the three steps involved in process migration.	C309.5	BTL-1
50	**What thread operations are the most significant in cost? (NOV/DEC 2017)** **Thread operations most significant in cost are:** **□ The overhead involved in creating a new process greater than creating a thread within a process.** **□ Switching between threads sharing the same address space is cheaper than switching between processes that have their own address space.** **□ Threads allow parallelism combined with sequential execution and blocking system calls.** **□ Resource sharing can be achieved more efficiently between threads of a process than between processes.**	C309.5	BTL-1
51	**What are Priority assignment policy? (NOV/DEC 2017)** **Selfish** **Altruistic** **Intermediate** **Migration Limiting Policies**	C309.5	BTL-1

52	**Draw the pictorial reperesentation of lifecycle of java thread. NOV/DEC 2018**  new blocked start acquiring a lock lock acquired runnable waiting for notification runout method notification acquired waiting terminated waiting for timeout notification timeout or notification occurred time waiting	C309.5	BTL-2
53	**What is User Mode Scheduling(UMS)? NOV/DEC 2018** User-mode scheduling (UMS) is a lightweight mechanism that applications can use to schedule their own threads. An application can switch between UMS threads in user mode without involving the system scheduler and regain control of the processor if a UMS thread blocks in the kernel.	C309.5	BTL-1
54	**Write the uses of threads.** • Threads minimize the context switching time. • Use of threads provides concurrency within a process. • Efficient communication. • It is more economical to create and context switch threads. • Threads allow utilization of multiprocessor architectures to a greater scale and efficiency.	C309.5	BTL-1
	PART B		
1	Discuss about the main functions of Process Management in a Distributed Environment. **Pradeep K Sinha-pg-no-390**	C309.5	BTL-6
2	Explain Process Migration with suitable example? **MAY/JUNE 16, NOV/DEC 2016 , ARPIL/MAY 2018 Pradeep K Sinha-pg-no-393**	C309.5	BTL-5
3	Write notes on desirable features of a good process migration mechanism. **Pradeep K Sinha-pg-no-397**	C309.5	BTL-5
4	Explain the steps involved in process migration. Write the advantages of Process migration **Pradeep K Sinha-pg-no-393**	C309.5	BTL-5
5	How process migration is done in heterogenous systems. **APRIL/MAY 2017 Pradeep K Sinha-pg-no-394**	C309.5	BTL-6

6	Explain about various Thread Models in detail. **NOV/DEC 2016, APRIL/MAY 2017 Pradeep K Sinha-pg-no-399**	C309.5	BTL-5
7	Explain the design Issues in Threads Package. How to Implement a Thread Package? **Pradeep K Sinha-pg-no-403**	C309.5	BTL-6
8	Write about Remote Execution of Threads. Pradeep K Sinha-pg-no-413	C309.5	BTL-5
9	Explain the desirable features of Global Scheduling **Algorithm Pradeep K Sinha-pg-no-348**	C309.5	BTL-5
10	Explain Task Assignment Approach. **MAY/JUNE 16 Pradeep K Sinha-pg-no-351**	C309.5	BTL-5
11	Discuss Load-balancing Approach. **MAY/JUNE 16**, **NOV/DEC 2016, APRIL/MAY 2017**, **APRIL/MAY 2018 Pradeep K Sinha-pg-no**-355	C309.5	BTL-5
12	Explain Load-sharing Approach. **MAY/JUNE 16, NOV/DEC 2016 Pradeep K Sinha-pg-no-367**	C309.5	BTL-5
13	Give the techniques and methodologies for scheduling process of a distributed systems. **APRIL/MAY 2017 Pradeep K Sinha-pg-no-348**	C309.5	BTL-5
14	i.state the issues in load balancing algorithms. ii.What is process migration?state the issues in migration, describe negotiation in migration process with neat sketch. **NOV/DEC 2018**	C309.5	BTL-5
15	i.Give a brief account on desired features of scheduling algorithms. ii.Define thred, elucidate the actions involved in multithreaded architecture and multithreaded models with appropriate sketch. **NOV/DEC 2018**	C309.5	BTL-5