ORAL QUESTIONS

Subject LP-5 Class-BE-IT A.Y 2022-23 (Sem-II)

RMI

1. What is RPC and LRPC?
2. What is the advantage of RPC 2 over RPC?
3. How do we provide security to RMI classes?
4. What are Layered Protocols?
5. What is Remote Method Invocation?
6. What is Distributed File System (DFS)?
7. What do you mean by Auto mounting?
8. What is the advantage of RPC2 over RPC?
9. What are advances in CODA as to AFS?
10. Which is the most Important Feature of CODA?

Lab Manual - Software Laboratory - V 15 VPKBIET, Baramati
Implementation of Simple Remote Method Invocation Using Java RMI

11. What are Stubs and Skeletons?
12. How communication does takes place in NFS?
13. Explain the Naming concept in NFS?
14. How Synchronization takes place in NFS?

15. How do you implement file locking in NFS?

16. What is the relationship between the RMI and CORBA?

What are the different types of classes that are used in RMI?

- The classes define the blueprint that consists of the methods and behavior of a particular structure.
- The classes are:

Remote class:
- It is the class that consists of the instances that can be accessed remotely from the

server and also consists of different properties that are required to be accessed by the client and server during request and response time.

Remote class object:
- Wherever this object is defined the instances of the class can be accessed using other object.
- On other computers that are not in network it can be accessed through the use of object handles.

Serializable class:
- It is the class that consists of instances that can be marshaled or turned in a linear sequence to represent the bits.

Serializable class objects:
- It allows the request to be transmitted from one computer to another.
- It allows the easy implementation of the serializable class.

What is the basic principle of RMI architecture ?

The RMI architecture is based on a very important principle which states that the definition of the behavior and the implementation of that behavior, are separate concepts. RMI allows the code that defines the behavior and the code that implements the behavior to remain separate and to run on separate JVMs.

What are the layers of RMI Architecture ?

The RMI architecture consists of the following layers:

- **Stub and Skeleton layer**: This layer lies just beneath the view of the developer. This layer is responsible for intercepting method calls made by the client to the interface and redirect these calls to a remote RMI Service.
- **Remote Reference Layer**: The second layer of the RMI architecture deals with the interpretation of references made from the client to the server’s remote objects. This layer interprets and manages references made from

clients to the remote service objects. The connection is a one-to-one (unicast) link.

- **Transport layer**: This layer is responsible for connecting the two JVM participating in the service. This layer is based on TCP/IP connections between machines in a network. It provides basic connectivity, as well as some firewall penetration strategies.

What is the role of the java.rmi.Naming Class ?

The java.rmi.Naming class provides methods for storing and obtaining references to remote objects in the remote object registry. Each method of the Naming class takes as one of its arguments a name that is a String in URL format.

What is the difference between using bind() and rebind() methods of Naming Class ?

The bind method bind is responsible for binding the specified name to a remote object, while the rebind method is responsible for rebinding the specified name to a new remote object. In case a binding exists for that name, the binding is replaced.

85. What are the steps involved to make work a RMI program ?

The following steps must be involved in order for a RMI program to work properly:

- Compilation of all source files.
- Generatation of the stubs using rmic.
- Start the rmiregistry.
- Start the RMIServer.
- Run the client program.

89. Explain Marshalling and demarshalling.

When an application wants to pass its memory objects across a network to another host or persist it to storage, the in-memory representation must be converted to a suitable format. This process is called marshalling and the revert operation is called demarshalling.

90. Explain Serialization and Deserialization.

Java provides a mechanism, called object serialization where an object can be represented as a sequence of bytes and includes the object's data, as well as information about the object’s type, and the types of data stored in the object. Thus, serialization can be seen as a way of flattening objects, in order to be stored on disk, and later, read back and reconstituted. Deserialisation is the reverse process of converting an object from its flattened state to a live object.

RPC

1. What is remote procedure call ?
2. What is Client Stub in remote procedure call ?Design a distributed application using Remote Procedure Call
3. What is Server Stub in remote procedure call?
4. What is the sequence of events during remote procedure call?
5. Explain the di_erence between remote procedure call and local calls?
6. Why remote procedure call (RPC) doesnt _t in OSI model?
7. What is LRPC ? In which case it is used?

CORBA

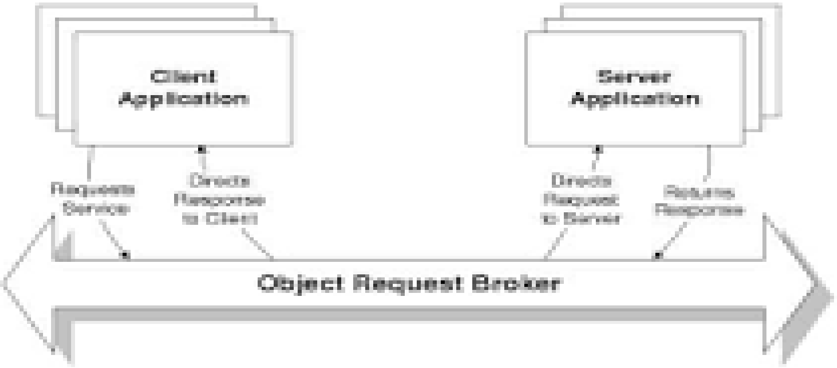
Question 8. What Is Corba? What Does It Do?

Answer :

CORBA is the acronym for Common Object Request Broker Architecture, OMG's open, vendor-independent architecture and infrastructure that computer applications use to work together over networks. Using the standard protocol IIOP, a CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network

CORBA Features

- The data that the object makes public.
- The operations that the object can respond to, including the complete signature of the operation. CORBA operations are mapped to Java methods, and the IDL operation parameter types map to Java datatypes.
- Exceptions that the object can throw.
    - What are the objectives of CORBA?



-
-
-
- The goal of CORBA is to promote an object-oriented approach to building and integrating distributed software applications. An object model for building distributed applications. A common set of application programming objects to be used by the client and server applications.

What is CORBA?

See also: What is Real-time CORBA?

The Common Object Request Broker Architecture (CORBA) is a standard developed by the Object Management Group (OMG) to provide interoperability among distributed objects. CORBA is the world's leading middleware solution enabling the exchange of information, independent of hardware platforms, programming languages, and operating systems. CORBA is essentially a design specification for an Object Request Broker (ORB), where an ORB provides the mechanism required for distributed objects to communicate with one another, whether locally or on remote devices, written in different languages, or at different locations on a network.

The CORBA Interface Definition Language, or IDL, allows the development of language and location-independent interfaces to distributed objects. Using CORBA, application components can communicate with one another no matter where they are located, or who has designed them. CORBA provides the location transparency to be able to execute these applications.

CORBA is often described as a "software bus" because it is a software-based communications interface through which objects are located and accessed. The illustration below identifies the primary components seen within a CORBA implementation.

- **Interoperability in distributed system** is the ability of different systems, devices, applications or products to connect and communicate in a coordinated way, without effort from the end user.

Message Passing Interface (MPI)
5.6 Viva Questions

What is MPI in distributed system?

The message passing interface (MPI) is a standardized means of exchanging messages between multiple computers running a parallel program across distributed memory. In parallel computing, multiple computers – or even multiple processor cores within the same computer – are called **nodes.**

1. What is my processor id number ?
2. How to get total number of processes created?
3. What is the parameter passed to the MPI Program while running?

4. MPI is Used for parallel computers, clusters, and heterogeneous networks as a message passing library

5. **What Is Marshalling And Unmarshalling?**
Answer :
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.

6. Which are the types of message passing?

- 2.1 Synchronous message passing.
- 2.2 Asynchronous message passing.
- 2.3 Hybrids.

Berkely Algorithm:-

Berkeley's Algorithm is a clock synchronization technique used in distributed systems. The algorithm assumes that each machine node in the network either doesn't have an accurate time source or doesn't possess a UTC server.

Algorithm

1) An individual node is chosen as the master node from a pool node in the network. This node is the main node in the network which acts as a master and the rest of the nodes act as slaves. The master node is chosen using an election process/leader election algorithm.

2) Master node periodically pings slaves nodes and fetches clock time at them using Cristian's algorithm.

The diagram below illustrates how the master sends requests to slave nodes.

3) Master node calculates the average time difference between all the clock times received and the clock time given by the master's system clock itself. This average time difference is added to the current time at the master's system clock and broadcasted over the network.

Pseudocode for above step:

features of Berkeley's Algorithm:

Centralized time coordinator: Berkeley's Algorithm uses a centralized time coordinator, which is responsible for maintaining the global time and distributing it to all the client machines.

Clock adjustment: The algorithm adjusts the clock of each client machine based on the difference between its local time and the time received from the time coordinator.

Average calculation: The algorithm calculates the average time difference between the client machines and the time coordinator to reduce the effect of any clock drift.

Fault tolerance: Berkeley's Algorithm is fault-tolerant, as it can handle failures in the network or the time coordinator by using backup time coordinators.

Accuracy: The algorithm provides accurate time synchronization across all the client machines, reducing the chances of errors due to time discrepancies.

Scalability: The algorithm is scalable, as it can handle a large number of client machines, and the time coordinator can be easily replicated to provide high availability.

Security: Berkeley's Algorithm provides security mechanisms such as authentication and encryption to protect the time information from unauthorized access or tampering.
The code below is a python script that can be used to trigger a master clock serve

What is Berkeley's algorithm in distributed system problems?

Berkeley's Algorithm is a distributed algorithm for computing the correct time in a network of computers. The algorithm is designed to work in a network where clocks may be running at slightly different rates, and some computers may experience intermittent communication failures.

Why clock synchronization is necessary in distributed system?

Clock synchronization is necessary for the ordering of events and to preserve the state of resources. As per algorithms, we can say that for clock synchronization there is need to consider propagation time of messages among each node in both types of algorithms centralized and distributed.

Other clock synchronization algorithms:
Cristian's Algorithm is a clock synchronization algorithm is used to synchronize time with a time server by client processes. This algorithm works well with low-latency networks where Round Trip Time is short as compared to accuracy while redundancy-prone distributed systems/applications do not go hand in hand with this algorithm. Here Round Trip Time refers to the time duration between the start of a Request and the end of the corresponding Response.
Below is an illustration imitating the working of Cristian's algorithm:

Token Ring with Mutual Exclusion

Distributed systems are computer networks in which a number of independent computers collaborate to accomplish a particular task.

Maintaining mutual exclusion among nodes, which makes sure that several nodes do not access a shared resource at the same time, is one of the most important issues in distributed systems

- Token Ring algorithm achieves mutual exclusion in a distributed system by creating a bus network of processes. A logical ring is constructed with these processes and each process is assigned a position in the ring. Each process knows who is next in line after itself.
- In this algorithm it is assumed that all the processes in the system are organized in a logical ring. The figure blow describes the structure.
- The ring positions may be allocated in numerical order of network addresses and is unidirectional in the sense that all messages are passed only in clockwise or anti-clockwise direction.
- When a process sends a request message to current coordinator and does not receive a reply within a fixed timeout, it assumes the coordinator has crashed. It then initializes the ring and process Pi is given a token.
- The token circulates around the ring. It is passed from process k to k+1 in point to point messages. When a process acquires the token from its neighbor it checks to see if it is attempting to enter a critical region. If so the process enters the region does all the execution and leaves the region. After it has exited it passes the token along the ring. It is not permitted to enter a second critical region using the same token.
- If a process is handed the token by its neighbor and is not interested in entering a critical region it just passes along. When no processes want to enter any critical regions the token just circulates at high speed around the ring.
- Only one process has the token at any instant so only one process can actually be in a critical region. Since the token circulates among the process in a well-defined order, starvation cannot occur.
- Once a process decides it wants to enter a critical region, at worst it will have to wait for every other process to enter and leave one critical region.
- The disadvantage is that if the token is lost it must be regenerated. But the detection of lost token is difficult. If the token is not received for a long time it might not be lost but is in use
-
- The algorithm works as follows:
- When the ring is initialized, process 0 is given a token. The token circulates around the ring. When a process acquires the token from its neighbor, it checks to see if it is attempting to enter a critical region. If so, the process

enters the region, does all the work it needs to, and leaves the region. After it has exited, it passes the token to the next process in the ring. It is not allowed to enter the critical region again using the same token. If a process is handed the token by its neighbor and is not interested in entering a critical region, it just passes the token along to the next process.

- This workbench displays a distributed system with six nodes, each node having a status flag. If the flag is green it means the node is ready to accept the request. The yellow flag means a request to enter a critical region is generated and the node is waiting for token. If the flag is red, the node is in a critical region. In the interactive mode of this workbench, you can generate requests for nodes by pushing corresponding Node buttons at the bottom. You can also execute the workbench in the continuous mode by pushing the Animate button. The animation speed can be varied using Speed scroll bar.

Which algorithm is used to handle mutual exclusion in distributed system?

Lamport's Algorithm
Lamport's Algorithm for Mutual Exclusion in Distributed System

What is token based algorithm?

How can we prevent deadlock in distributed system?

Just like in centralized deadlock prevention, in distributed deadlock prevention approach, a transaction should acquire all the locks before starting to execute. This prevents deadlocks. The site where the transaction enters is designated as the controlling site.

Election algorithm:-

What is the election algorithm in distributed system types?
Election algorithm assumes that every active process in the system has a unique priority number. The process with highest priority will be chosen as a new coordinator. Hence, when a coordinator fails, this algorithm elects that active process which has highest priority number.

How many types of messages are there in election algorithm?

The algorithm uses the following message types: Election Message: Sent to announce election. Answer (Alive) Message: Responds to the Election message. Coordinator (Victory) Message: Sent by winner of the election to announce victory.

What is leader election algorithm and why do we need this algorithm?

A leader election algorithm guides a cluster to jointly agree on one node to serve as a leader with as few back-and-forth interactions as possible. Generally, the algorithm assigns each node one of three states: Leader, Follower, or Candidate.

What is the difference between ring and bully election algorithm?

The bully algorithm is more robust than the ring-based algorithm because it can continue even if a process crashes during the election. In order to detect such failures the system must be assumed to be synchronous: timeouts are used.

What is the bully election algorithm?

Bully algorithm mechanism

The algorithm operates by identifying all the non-faulty nodes and electing the node with the largest identifier as the leader. There can be three kinds of messages that nodes would exchange between each other during the bully algorithm: Election message. OK message.

What is called election?

An election is a formal group decision-making process by which a population chooses an individual or multiple individuals to hold public office.

What is quorum based algorithm?

A quorum is the minimum number of votes that a distributed transaction has to obtain in order to be allowed to perform an operation in a distributed system. A quorum-based technique is implemented to enforce consistent operation in a distributed system.

What is the time complexity of bully algo?
This algorithm runs in O(n2) time in the worst case when lowest ID process initiates the election. The name bully is given to the algorithm because the higher ID processes are bullying the lower ID processes to drop out of the election.

Web service

Define Web Service?

A web service is a kind of software that is accessible on the Internet. It makes use of the XML messaging system and offers an easy to understand, interface for the end users.

3) Give me an example of real web service?

One example of web services is IBM Web Services browser. You can get it from IBM Alphaworks site. This browser shows various demos related to web services. Basically web services can be used with the help of SOAP, WSDL, and UDDI . All these, provide a plug-and-play interface for using web services such as stock-quote service, a traffic-report service, weather service etc.

4) How you define web service protocol stack?

It is basically set of various protocols that can be used to explore and execute web services. The entire stack has four layers i.e. Service Transport, XML Messaging, Service Description and Service Discovery.

6) Define XML – RPC?

It is a protocol that makes use of XML messages to do Remote Procedure Calls.

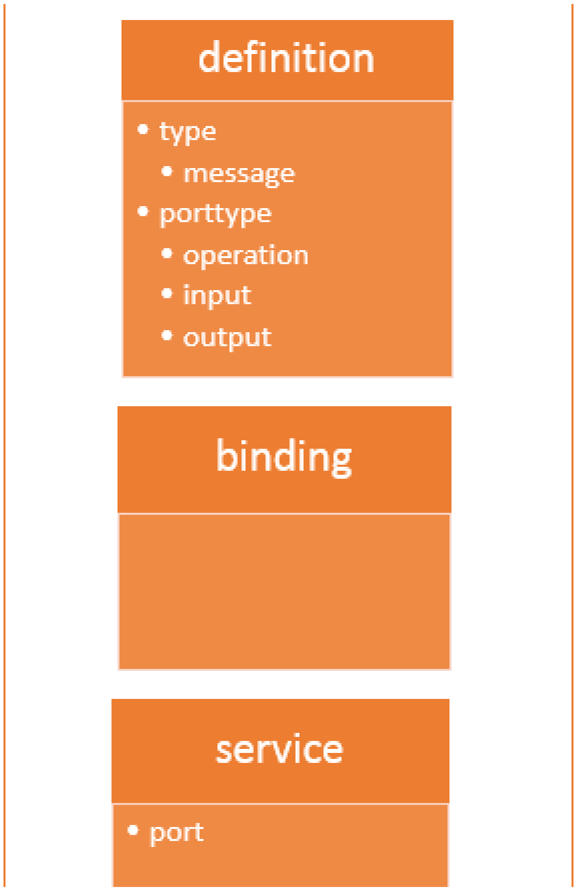
7) Define SOAP?

SOAP is an XML based protocol to transfer between computers.

8) Define WSDL?

It means Web Services Description Language. It is basically the service description layer in the web service protocol stock. The Service Description layer describes the user interface to a web service.

Structure of a WSDL



9) What kind of security is needed for web services?

The security level for web services should be more than that of what we say Secure Socket Layer (SSL). This level of security can be only achieved from Entrust Secure Transaction Platform. Web services need this level of security to ensure reliable transactions and secure confidential information.

10) What UDDI means?

UDDI stands for Universal, Description, Discovery, and Integration. It is the discovery layer in the web services protocol stack.

11) What tools are used to test a web service?

I have used SoapUI for SOAP WS and Firefox poster plugin for RESTFul Services.

12) What is REST?

REST stands for Representational State Transfer. REST itself is not a standard, while it uses various standards such as HTTP, URL, XML/HTML/GIF/JPEG (Resource Representations) and text/xml, text/html, image/gif, image/jpeg, etc (MIME Types).

13) Is there any special application required to access web service?

No, you don't need to install any special application to access web service. You can access web service from any application that supports XML based object request and response.

14) Can you name few free and commercial implementations for web services?

The implementations I know are Apache SOAP, JAX-WS Reference Implementation, JAX-RS Reference Implementation, Metro, Apache CXF, MS.NET and Java 6.

15) Differentiate between web services, CORBA and DCOM?

Web services transfer/receive messages to/from application respectively, via HTTP protocol. It uses XML to encode data.

CORBA and DCOM transfer/receive messages to/from application respectively, via non-standard protocols such as IIOP and RPC.

30) Can you tell few benefits of web services?

The biggest advantage of web service is that is supported by wide variety of platforms. Moreover, in near future, web services may spread its boundary and enhance new methods that will provide ease to clients. The enhancement will not affect the clients, even if they offer old methods and parameters.

33) Explain in brief, what UDDI is?

UDDI (Universal Description, Discovery, and Integration) provides consolidated directory for web services on the internet. Clients use UDDI to find web services as per their business needs. It basically hosts the web services from various companies. In order to share web services, you need to publish it in UDDI.

34) Explain the .NET web services supported data types?

.Net web services uses XML-based standards to transfer/receive information. Thus, .NET web services can only works with data types known by XML schema standard. Like FileSteam, Eventlog etc. are not recognized by the XML schema standards and hence, not supported in web services.

35) How a .NET web service is tested?

ASP.NET uses a test page routinely, when one calls for the URL of .asmx file in any browser. This page shows complete information regarding web services.

36) How a .NET web service is consumed?

Since we know that web services are constructed on XML standards. Therefore, clients need to have complete understanding of XML-based messages to interchange messages. Clients can communicate with web services through .NET framework that offers proxy mechanisms. These proxy mechanisms have detailed information regarding data sharing within web services that can be easily used by the clients.

37) Can you name the two Microsoft solutions for distributed applications?

The two Microsoft solutions for distributed applications are .NET Web Services and .NET Remoting

General Questions in Distributed systems Other than Practical Assignments

Q.What Are Distributed Systems? ...
Q.What are distributed systems used for? ...
Q.How does a distributed system work? ...

Q.What has distributed garbage collection? ...
Q.What have distributed computing techniques? ...
Q.What have distributed algorithms? ...
Q.What are the challenges in designing a distributed system? ...
Q. What is the Difference between Networked System and Distributed System?
Q. Name few characteristics of Distributed Systems?
Q. Name Some Case Studies of Distributed Systems which you have studied?
Q. If you are said to design a Distributed Systems for your Client which design issues you are going to consider?

Q. Give Some examples of a distributed system?

Following are the main example of distributed systems:

1. **The telephone system and cellular networks** are excellent examples of a Distributed System.
2. **The World Wide Web** is another distributed system that allows people to connect with each other through computers all over the world which creates a virtual community where anything can happen.
3. **Parallel computation**
4. **Aircraft control systems** employ similar principles by using information from different sources at once in order for planes not only to know what direction they're going but also how fast or slow they should fly according to wind conditions etc…
5. **Peer-to-peer networking** apps like BitTorrent (for downloading)

Q Transperancies in DS
Q.Middleware in DS

TCP & UDP

7. Explain the TCP and UDP Protocols
8. What is a Distributed Systems?
9. Give few examples of distributed systems?
    10. What is the Difference between Networked System and Distributed System?
    11. Name few characteristics of Distributed Systems?
    12. Name Some Case Studies of Distributed Systems which you have studied?

13. If you are said to design a Distributed Systems for your Client which design issues you are going to consider?
14. Explain the TCP and UDP Protocols
15. What are Different challenges faced by Distributed Systems?
16. Name Popular System Models in Distributed Systems?
17. Explain the Difference between Message oriented Communication and stream Ori-ented Communication.
18. What are Layered Protocols?

HDFS(Hadoop Distributed File System)

Viva Questions
1. Explain HDFS.
2. Explain about the basic parameters of mapper and reducer function.
3. Which are the various components of Hadoop Architecture?
4. How data is spilt in Hadoop?
5. When is it not recommended to use MapReduce paradigm for large scale data processing?
6. What is the fundamental di_erence between a MapReduce Split and a HDFS block?
7. What is Big Data? Which are the various methods for analyzing Big Data?
8. Discuss the various challenges which need to faced while analyzing Big Data.
9. Compare RDBMS with Hadoop MapReduce.
10. List the con_guration parameters that have to be speci_ed when running a MapReduce job.
11. What is the relationship between Job and Task in Hadoop?