# Section 9: Distributed Processing

## Question 1

Describe Synchronous messaging.

### Answer

Synchronous messaging describes a client-server protocol and implies synchronized client and server processes. That is when the client makes its request of the server, the client blocks its execution until a response is received.

In practical terms, the client’s request should be time limited and a “timeout” error (or exception) will be generated if the server’s response is not received.

## Question 2 (20 Points)

1. Describe how information flows between two endpoints in a TCP Socket.
2. Describe the relationship between InputStream and OutputStream with respect to the flow of information.
3. How is blocking I/O of an InputStream used to regulate this flow?

### Answer

A TCP Socket represents a two-way communication channel between two processes. Each end of this channel has two endpoints that allow the processes to exchange information i.e. streams of bytes. Each endpoint allows a process to both receive bytes from, and to send bytes to the other process.

To accomplish this each endpoint is supported by an InputStream and OutputStream. The InputStream has a read() operation that is used to read the bytes sent by the other process. The OutputStream has a write() operation that is used to send bytes to the other process.

The InputStream of each endpoint is blocking. That is, when a process read()’s from its InputStream, the process blocks if the stream is empty and will continue to block until the other process writes data into it’s OutputStream.

## Question 3

Describe the purpose of the two protocols in TCP / IP.

### Answer

IP or Internet Protocol is responsible for transporting fixed length packets of data (bytes) from one machine to another machine across the IP network. The IP network supports the ‘routing’ of packets between connected machines across the world-wide Internet.

TCP or Transport Control Protocol is responsible for implementing the socket mechanism by breaking the stream of bytes transmitted by a process on its socket endpoint’s OutputStream into many separate packets that are individually routed (routers) on the IP network to the destination machine. TCP is also responsible for re-assembling the packets in the correct order to reproduce the original stream of bytes on the InputStream side of the receiving process.

## Question 4

Describe in detail the five steps of the two-phase message protocol between client and server processes. Be sure to include how blocking I/O enter into the implementation.

### Answer

The Client-Server protocol defines the exchange of two messages (request and response) between two processes (client and server).

1. The server process is started and listens for connection requests from any client.
2. The client process contacts / connects to the server process establishing a TCP/IP socket. Once established the Server Process waits for the client’s request message.
3. The Client Process builds and sends a request message to the Server Process over the TCP/IP socket. Once the request message is sent, the client process waits for the server to send its response message.
4. The server process receives and processes the request message in some application-specific manner based on the operation requested by the client. In response to this processing, the server generates a response message which is sent back to the client over the same socket.
5. The client process receives and processes the response message and continues its execution.

The server waits for the client’s message to arrive by blocking on a read() of its TCP socket endpoint. The server process’s read() on its InputStream blocks until the client sends a message across the socket. Conversely, after the client sends its request message, the client reads and blocks on its endpoint’s InputStream until the server sends its response message.

## Question 5

What are the names and purposes of each of the tiers in a Three-Tier Architecture?

### Answer

**Presentation Tier**: The presentation tier contains the software and services that presents information to, and gathers information from, the system’s users. This may be an application running on a phone or on a browser.

**Service Tier**: The service tier contains the software and services that implement the application information processing and business rules. The service tier software architecture is typically made up of Controller (see the design pattern) and services that are invoked by client requests from the presentation tier.

**Data Tier**: The data tier contains the software and services responsible for persisting data in databases. The persistence of information is often complex and tightly coupled to the specific database and schema. So it is wise to encapsulate the implementation of persistence from the services in the service tier.

It is common for the components / processes that make up each tier to run on separate machines, sometimes using a cluster to increase capacity. These processes communicate with each other using Remote Procedure Calls or other distributed processing technology.

## Question 6

1. Describe the two-part structure of a ‘Request Message’ as described in the slides.
2. What is the purpose of the first part / section of the message?

### Answer

1. The request message contains two types of information. The first part is the operation ID and the optional second, third parts are arguments needed by the server process to execute the required operation.
2. The operation ID is a string or integer that uniquely identifies the operation type that is being requested by the client. The client and server processes must agree on (be designed with) the values that can be used as an operation ID. For example, the slides describe a server with Op IDs of “echo” and “reverse”.

## Question 7

What two pieces of information are needed by a client process to contact (connect to) a listening server process using TCP/IP?

### Answer

The client must ‘know’ the IP address of the machine the server process is running on, and port number that the server process listening on for incoming connection requests.

## Question 8

How does server clusters promote performance scaling of the system?

### Answer

The client’s requests are distributed evenly among M servers allowing M requests to be processed concurrently. This increases the performance of the system M-fold. This is the theory anyway, because the actual scaling in performance depends on how much of the request processing can be executed concurrently i.e. see Amdahl’s rule.

## Question 9

How does server clusters promote high-availability of the system?

### Answer

If any of the cluster’s servers crashes, the load balancer will detect the failure and remove the failed server from the cluster i.e. stop forwarding client requests. High availability means that the remaining M-1 servers remain operational the service itself remains available to the clients.

## Question 10

Describe the role the Load Balancer plays in implementing a scaling and high-availability in a cluster of processors (servers).

### Answer

The Load Balancer is a device (hardware or software) that is attached to each processor in the cluster. The LB receives all client requests and forwards each (request) to one of the clustered processors. The LB achieves scaling by forwarding N client-requests evenly across M processors. The LB achieves high-availability by detecting when one of its processors has died (stopped responding) and can remove the broken processor from its processor pool.

It is important to note that all of the cluster’s processors is running the same services (software), so any server is capable of processing the client’s requests.