# Section 9: Distributed Processing

## Question 1

Describe the difference between Synchronous and Asynchronous messaging.

### Answer

Synchronous messaging almost always describes a client-server architecture and implies a synchronization between the 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.

Asynchronous messaging describes a message-driven (event-driven) architecture. Interacting processes have a peer-to-peer relationship, are not synchronized, and each process is capable of sending and receiving messages.

## Question 2

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 responsibility of the two layers in the TCP/IP protocol.

### 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 routed 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

1. Describe the two-phase message protocol that exists between client and server processes.
2. How does blocking I/O enter into the implementation of the client-server protocol?

### Answer

The Client-Server protocol defines the exchange of two messages between two (client & server) processes. The protocol starts with the Client Process sending a request message to the Server Process. This is accomplished by using a TCP/IP socket. The Server Process waits for the client’s message and ‘processes’ the message in some application-specific manner. This request message processing produces a result or response to the client’s message. The result is returned to the client by the server using the same TCP/IP Socket.

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

Fill in the blanks:

1. The business process describes the <A> though the enterprise
2. A business process is made up of <B> that accepts information as input, process the information, and generates results.

### Answer

1. flow of information
2. processing steps

## Question 7

1. Describe the role a Producer plays in implementing a message-based architecture and its relationship to message destinations.
2. Describe the role a Consumer plays in implementing a message-based architecture and its relationship to message destinations.
3. T|F: A component is either a Producer or a Consumer, but never both.

### Answer

Producers produce messages. The producer will process and produce a result that is used by different consuming component. The producer will deliver its messages to a message-type specific destination. The destination will forward the messages to consumers that have registered interested in messages of specific types.

Consumers consume (accept and process) messages. The consumer will accept the message and execute some application-specific code with the information provided by the message.

Flase: Many (most) message processing processes are both message consumers (accepting and processing a message) and produce messages as a result of their message processing behavior.

## Question 8

1. Select the option that makes this statement true: Asynchronous messaging systems are <more> <less> coupled that synchronous messaging systems.
2. Describe why Client-Server is considered strongly coupled.
3. Describe whey Messaging is considered weakly coupled.

### Answer

Asynchronous messaging systems are less coupled that synchronous messaging systems.

A synchronous messaging system is made up of clients that require a reference to a server in order to invoke an action on the server and (usually) receive a result / response. This reference is a pointer variable in a language or the network address of the machine hosting the server process.

An asynchronous messaging system is made up of message producers and consumers. Neither of these components is directly coupled to the other. Producers produce messages. Consumers consume messages. They share a common medium (a message server) which routes messages from producers to consumers. However consumers have no knowledge of the entity creating their inputs and producers have no knowledge of the entity consuming the information they produce. In most cases, the producer will continue to produce messages even when there are no consumers present in the system.

## Question 9

1. What are the two types of message destinations supported by most message server?
2. How does one destination differ from the other in terms of how a message is delivered to a set of registered consumers?

### Answer

**Message Queue**: A queue is a named destination that when shared by two or more consumers will route a message to exactly one of the consumers. Queues are point-to-point destinations.

**Message Topics**: A topic is a named destination that when shared by two or more consumers will route a message to every one of the consumers. Topics are broadcast destinations.

## Question 10

1. Describe the role the Load Balancer plays in implementing a cluster of web servers.
2. How does the Load Balancer promote performance scaling?
3. How does the Load Balancer support high-availability?

### Answer

The Load Balancer is a device (hardware or software) that is attached to each server in the cluster. The LB receives all client requests and forwards each (request) to one of the clustered servers. It is important to note that all of the cluster’s servers is running the same services (software), so any server is capable of processing the client’s requests.

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.

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.