

1.

- a) Suggest a design for a notification mailbox service that is intended to store notifications on behalf of multiple subscribers, allowing subscribers to specify when they require notifications to be delivered. Explain how subscribers that are not always active can make use of the service you describe. How will the service deal with subscribers that crash while they have delivery turned on? page 261

My design is: There is one interface Producer on the server side, and one interface Consumer on the client side. And activemq is the middleware for receiving and sending the messages.

```
public interface Producer {  
    public void publish() throws JMSException;  
}
```

```
public interface Consumer {  
    public void subscribe() throws JMSException, IOException;  
    public void unsubscribe() throws JMSException, IOException;  
}
```

client call subscribe() method to subscribe the service, and then the client will connect to the middleware activemq.

When the server has new mail, it can call the publish() method to connect and send the message to the middleware activemq. The middleware will send the message to all the clients that subscribe to the service.

Client calls the unsubscribe() method to unsubscribe the service, the client will disconnect from the middleware and no longer receive messages from the server.

Activemq has a mode of durable subscription, which can handle client failures. If the client crashes, the middleware records the transmission failure and client id and tries to send the message later. And when the clients restart, the client can receive the missing messages during its crash.

- b) Consider the version of the FireAlarm program written in JMS (Section 6.4.3). How is connecting to a named channel in JGroups achieved here? page 277

Connecting to a named channel in JGroups is achieved using JNDI(the Java Naming and Directory Interface). First it creates an empty context object, then it creates a connection(TopicConnection) between a client and the server through a connection factory. And then it calls the lookup() method in the context object to find the alarms topic.

The connection can be used to create sessions, which can support creating transactions. The session object supports methods for creating, producing and consuming messages.

c) Perform a comparative study between message passing and DSM.page 280

DSM	Message passing
uses shared memory, so it does not need separate communication operations;	Uses private address spaces, It allows processes to communicate while being protected;
it does not need marshal	Needs marshalling to communicate with other machines;
communication may execute with non-overlapping lifetimes;	must execute at the same time;
all remote data accesses are explicit	any read or update may or may not involve communication

2. implement a simple JMS Queue which can be useful for the notification mailbox discussed in Question 6.8 above. See the reference provided. [30 points]

Why do we need MyListener and MyReceiver differently in JMS Queue?

Implementation is in the homework4-2.jar, and there is a readMe.txt file to illustrate how to run the program.

I use activemq as middleware for the service. The producer is as the MySender, and the consumer is as the MyListener, and the middleware is as the MyReceiver. MySender sends the messages to the middleware, MyListener listens to the middleware to check if there are new messages, and MyReceiver receives the messages from the producer and sends them to the consumers.

3. See the Coding Tutorial PDF provided, and the below references. Do your own study of RMI Java examples to implement this. Implement a Java RMI Application in which the Client object is sending a list of 10 integers to the Server, and a remote method ['sort()'] o the server returns a sorted version of the same list back to the Client. [40 points]

Implementation is in the homework4-3.jar, and there is a readMe-3.txt file to illustrate how to run the program.