Unit-5

classmate
Date 23/11/16
Page

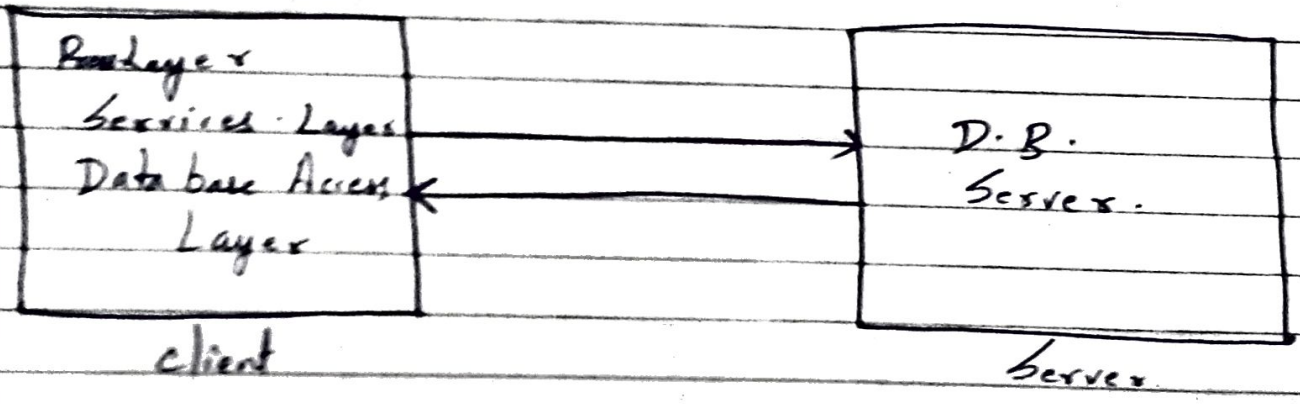
★ Different Types of Tier Architecture

Tier: Physical seperation of componente are called tier

→ 2 tier
→ 3 tier
→ N-tier

⇒ 2-tier:

- It also called client-server component/architecture.
- In 2 tier architecture server is the database server
- In such physical and data access layer, run in one machine which is nothing but client machine.



2 tier is divided into 2 parts 1) client application
2) database.

On client application site the code is written for saving the data in the SQL server database. Client sends the request to server and it process the request and send back with data.
The main problem of 2-tier architecture is the server can not respond multiple request at the same time

as a result it causes a data integrity issues.

Advantages:

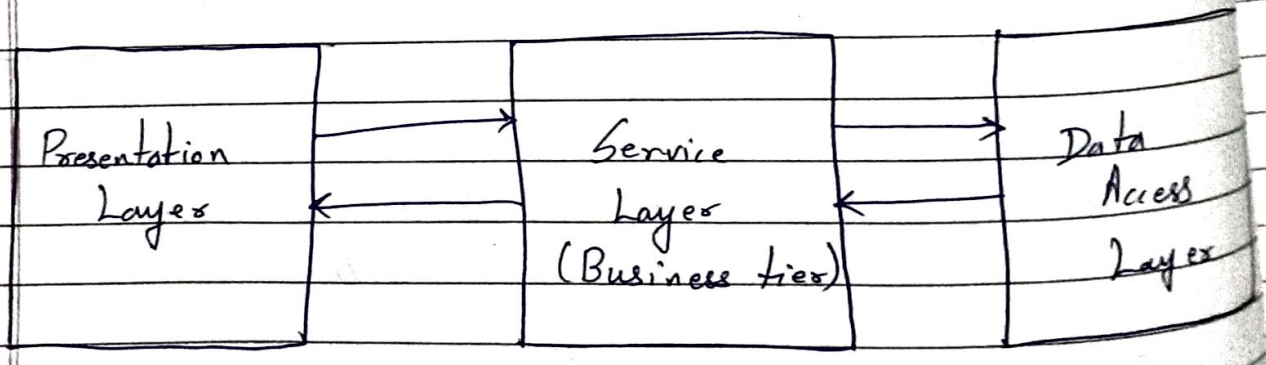
1) Easy to maintain and modification is bit easy.
2) Communication is faster.

Disadvantages:

1) In 2-tier architecture application performance will be degra upon increasing the users.
2) Cost ineffective

★ 3-tier Architecture

1) Physical seperation of service layer from presentation layer
2) One machine for service layer and another machine for presentation layer.
3) 3-tier architecture eliminates client side maintanance.



3-tier.

Presentation layer used for design purpose

Service layer

It act as a intermediate between presentation layer

and D.A.L.

2) It helps to communicate faster between client and data layer.

D.A.L (Data Access Layer)

- Act as database, used for data connection and to perform insert, update, delete, get data from database on our input data.

★ N-tier Architecture

1) Presentation:
In a typical web application, a browser running on a client machine handles presentation.

2) Dynamically generated presentation:
It supports web server using JSP, servlet, xml, XSL.

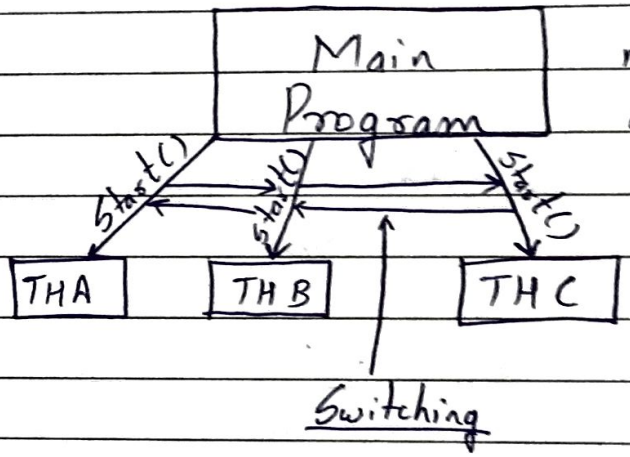
3) Bussiness Logic:
It is implemented in session EJB's.

4) Data Access:
It is implemented in Entity EJB and using JDBC.

★ Threads

Treads in java are subprograms of a main program and share the same memory space, they are also known as light weight process.

Since all threads are running on a single processor, The flow of execution is shared between the threads. The java interpreter handles the switching of control between the threads in such a way that it appears they are running concurrently.



main module of a program.

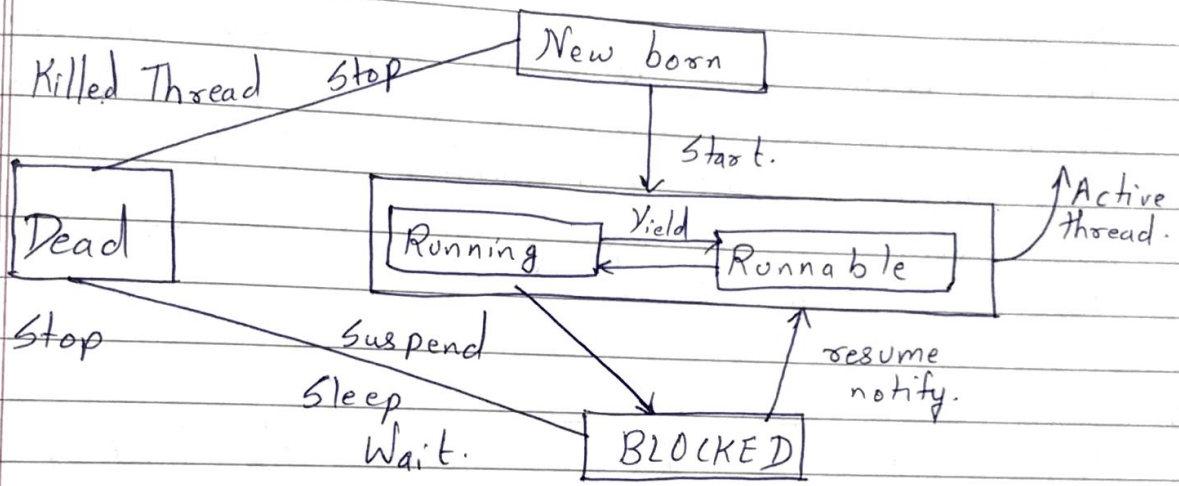
Threads can be implemented in two ways.

1) extends Thread
2) implement Runnable Interface.

Example:

```
Class A extends Thread
{
Public void run()
{
System.out.println("I am a Thread");
}
}
class B
{
```

```
P.S.V.M (String s[])
{
A a = new A();
a.start()
}
}
```



Difference between Throw & Throws

Throw	Throws.
1) Throw keyword is used to throw exception explicitly.	Throws clause is used to declare an exception.
2) Throws is followed by an instance variable	Throws is followed by exception class name.
3) Throw keyword is used inside the method body.	Throws clause is used in method declaration (Signature)
4) By using throw keyword we can't throw more than one exception.	By using throws clause multiple exception can be declare.
5) eg: throw new ArithmeticException ("an integer"); throw new IOException ("Connection failed");	eg: throws IOException, ArithmeticException, NullPointerException, ArrayIndexOutofBoundException.

★ ~~Thick~~

★ Thin Client:

Thin client is designed to be specially small so that the bulk of data processing occurs on the server. They act as a simple terminal to the server and require constant communication with the server.

1) Easy to develop as they require no extra or specialized software.
2) Needs to validate with the server after data capture.
3) If the server goes down, data collection is halted as the client needs constant communication with the server.
4) Client runs only and exactly as specified by the server.
5) Reduced security threat.

★ Thick Client: (fat client)

Thick client is one that will perform the bulk of the processing in client server applications. There is no need for continuous server communication as it mainly communicates archival ~~archivable~~ storage information to the server.

1) Data verified by client not server.
2) Require more resources but less servers.
3) Can store local files and applications.
4) Reduced server demands.
5) Increased security issues.

PrintWriter

PrintWriter is a character ~~string~~ stream class.

- We can use printWriter to write character based information such as character array and string to the response where as we can use servlet output stream to write byte array data to the response.

- We can use "servlet_response.get_writer()" to get the printwriter instance where as we can use servlet response.getoutputstream method to get the servletoutputstream object reference.

ServletOutputStream

whereas ServletOutputStream is a byte stream class.