

Multi Threading

Multi Threading is the concept used in Java based on Multi tasking. Multi Threading is the technique used to perform many things simultaneously. By using multi threading, we can make CPU more busy and hence reduce the idle time of CPU.

Java Thread Model:

This model is also similar to process based multi tasking. A Thread is a separate path of execution in a program. In Multi Threading model of multi tasking all thread have same address and are called light weight components.

Thread Priorities: Thread priorities are used by thread scheduler to decide when each thread should be allowed to run. These are the integer values.

There are three types of Thread priorities:

They are :

MIN_PRIORITY

MAX_PRIORITY

NORM_PRIORITY

MIN_PRIORITY.....VALUE IS 1

MAX_PRIORITY.....VALUE IS 10

NORM_PRIORITY.....VALUE IS 5

Synchronization:

When two or more threads need access to a shared resource, they need some way to ensure that the resource will be used by only one thread at a time. This is called

Synchronization.

By using this, the code is safe and the resource is not corrupted.

But the draw back is the execution is little bit slow.

Creating a thread:

We can create our own thread by extending a Thread class or implementing a Runnable interface.

Important methods in Thread class:

String getName(): Obtain thread's name

int getPriority(): Obtain a thread's priority.

boolean isAlive() : Determine if a thread is still running or not

static void sleep(long milliseconds): suspend a thread for a period of time.

public void start() : Start a thread by calling it's run method.

Imp. methods in Runnable interface;

public void run() :this method must be implemented by any class that implements this interface.

Creating Multiple Threads:

We can create multiple threads either by using a Thread class or a Runnable interface.

Eg:

Creating three user defined threads Thread1,Thread2 and Thread3 by extending a Thread class;

```
//creation of multiple threads
```

```
class Thread1 extends Thread
```

```
{
```

```
class Thread2 extends Thread
```

```
{
```

```
class Thread3 extends Thread
```

```
{
```

```
class Threaddemo
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
Thread1 t1=new Thread1();  
Thread1 t2=new Thread2();  
Thread1 t3=new Thread3();  
}  
}
```

Life cycle of a thread: In multithreading a thread is a separate path of execution in a program. Every thread has 5 life cycle stages. They are:-

1. New born state
2. Runnable state
3. Running state
4. Blocked state
5. Dead state

A thread is in one of these five states .It can move from one state to other state.

New born state: When a thread object is created the thread is born and is said to be in new born state.

Runnable state: in this state, the thread is ready for execution and is waiting for the availability of the processor.

Running state: Running means that the processor has given its time to the thread for its execution.

Blocked state: A thread is blocked when it is prevented from entering into the runnable state and subsequently in the running state. This happens when the thread is suspended, sleeping or waiting in order to satisfy certain requirements.

Dead state: it is the last stage of a thread .A running thread ends its life when it has completed executing its run () method. It is a natural death.

Inter-thread Communication:

Inter thread communication in Java is done by means of three methods.

They are wait(),notify() and notifyAll() methods.These methods are called only from a synchronized method.