Threads

Typically we can define threads as sub-process with lightweight with the smallest unit of process and also have separate path of execution.These threads are shared memory but they act independently.Hence,if there is an execution in thread that do not affect the working of other threads despite them sharing the same memory.

Multitasking:

To help of user operating system accommodates user the privilege of multitasking where user can perform multiple task simultaneously on same system.

This can be done in two ways:

1. Process-based Multitasking(Multi-processing): In this type of multitasking, process are heavy-weight and each process are allocated by separate memory area. As the process is heavy-weight the cost of communication bw process is high, and it takes long time for switching bw processes.

2. Thread-based Multitasking: As we discuss above, thread are provided with a lightweight nature and share the same address-space and the cost of communication bw the thread is also low.

Why threads are used?

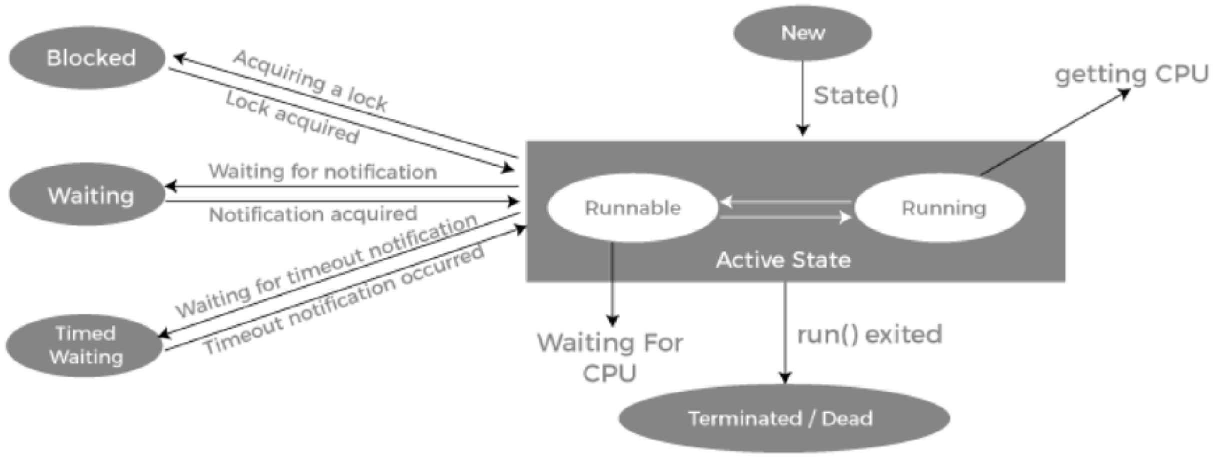
We can understand why threads are being used as they have the advantage of being light-weight and can provide communication bw multi-threads at a low-cost contribution affecting multitasking within the share memory environment.

What is segmentation in programming language?

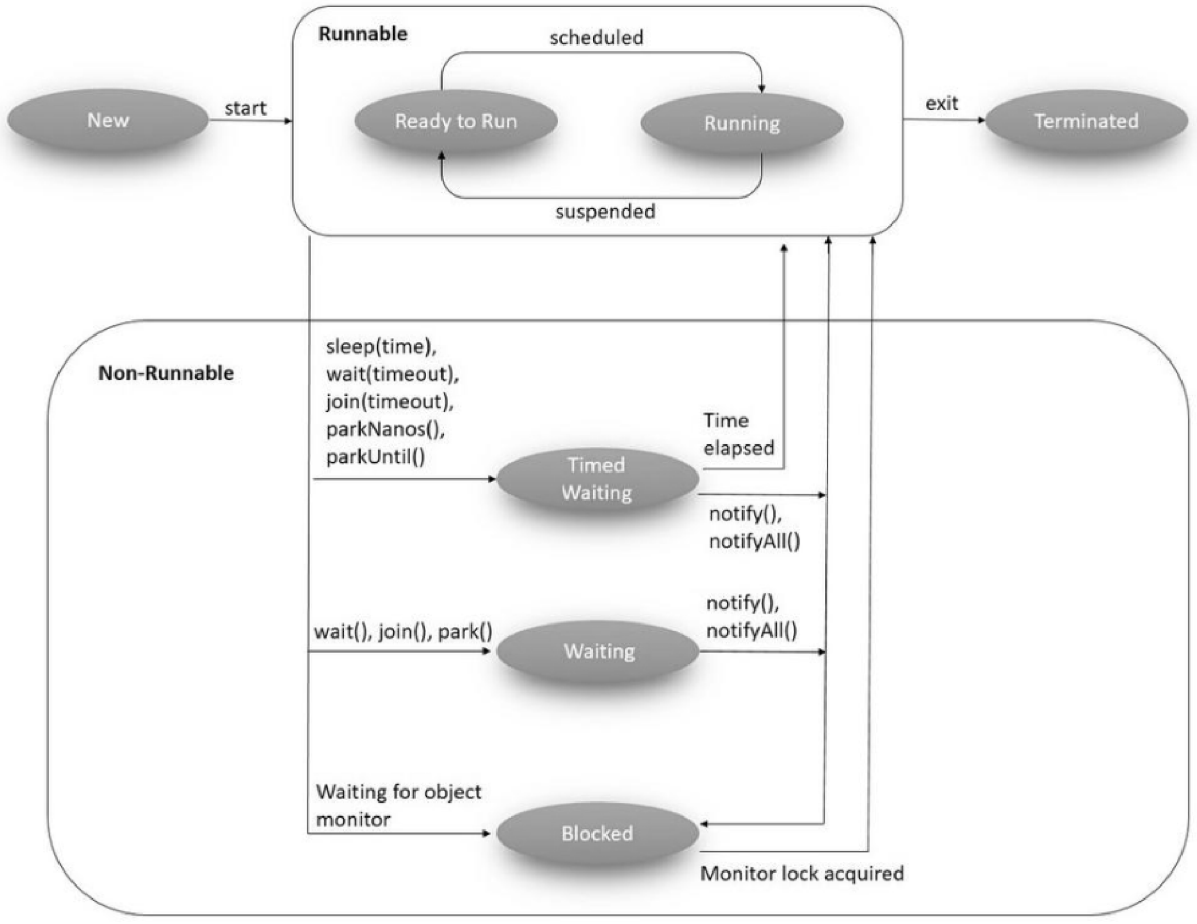
Segmentation is a memory management technique that can be used to improve the performance of an operating system. By dividing physical memory into segments, the operating system can better manage its memory use, improving efficiency and performance.

Life Cycle of Thread:

There are diff states, Thread Transfer into during its lifetime, Let us know about those states.



Life Cycle of a Thread



1. New State: Thread t1 = new Thread();

2. Active State: t1.start();

A thread that is new state by default get transfer to active state when it involves the start method.Internally, start method calls the run method.

3. Waiting State: t1.wait();

With the help of wait() method,we can wait our thread by invoking sleep() method.sleep() method can stop execution for specific time period.

4. Time waiting State: Will be done to stop the execution for some time period.sleep() method is invoked and after the time expire the thread starts execution its task.

5. Terminated State: A thread will be terminated in foll. conditions:

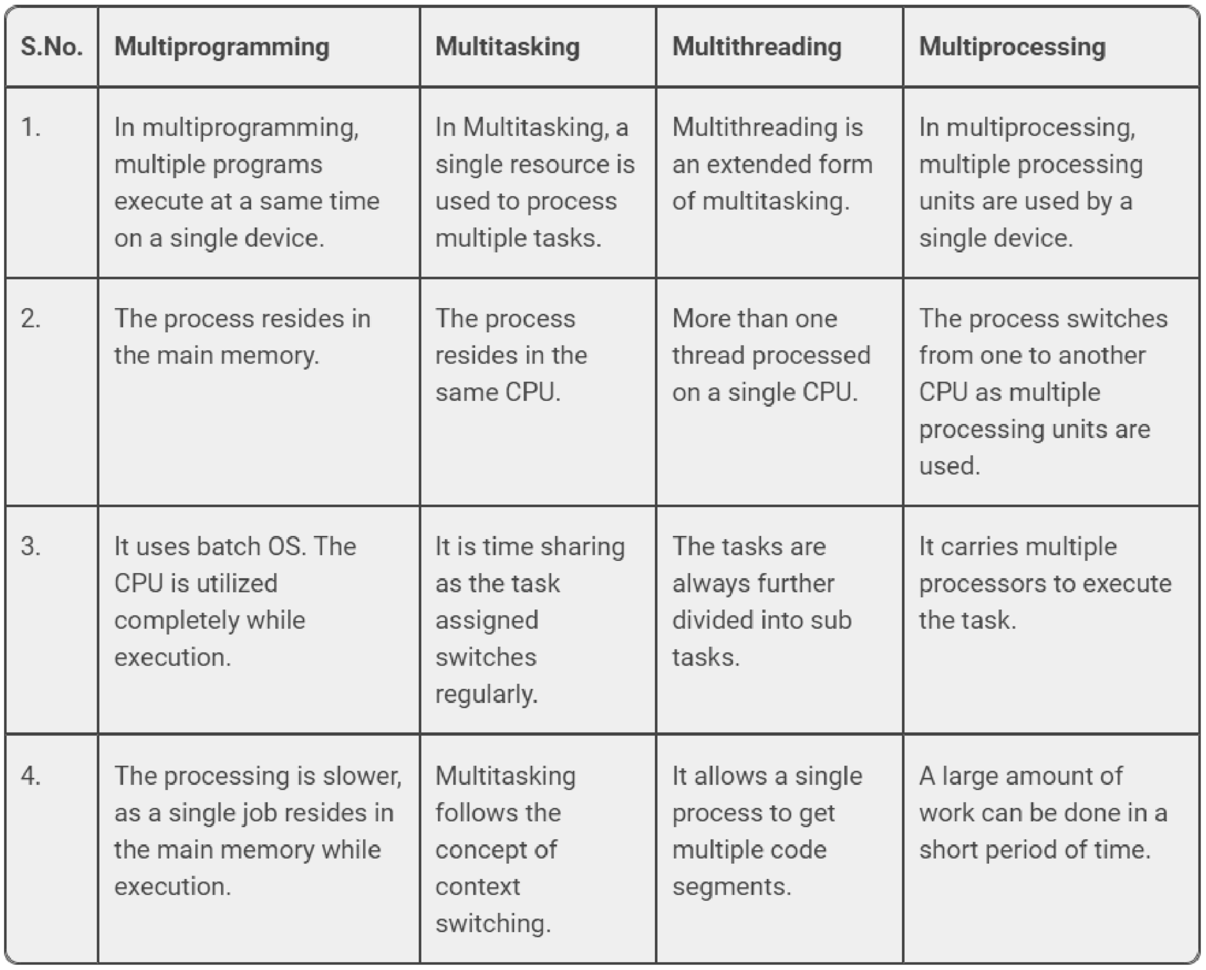
1. When the task finish normally.
2. Sometime thread may terminate due to abnormal events like segmentation fault,exceptions etc. and such type of termination is called abnormal events.
3. The terminate thread means it is dead and no longer available.

What is main thread?

We create main method in each and every java program which acts as entry pt. for the code to execute.Similarly, in this multi-threading concept each program has one main

thread which was provided default by jvm.
Hence,whenever program is created in java, jvm provides the main thread for its execution.

What is diff bw multi-tasking,multi-programming,multi-processing and multi-threading?



S.No.	Multiprogramming	Multitasking	Multithreading	Multiprocessing
1.	In multiprogramming, multiple programs execute at a same time on a single device.	In Multitasking, a single resource is used to process multiple tasks.	Multithreading is an extended form of multitasking.	In multiprocessing, multiple processing units are used by a single device.
2.	The process resides in the main memory.	The process resides in the same CPU.	More than one thread processed on a single CPU.	The process switches from one to another CPU as multiple processing units are used.
3.	It uses batch OS. The CPU is utilized completely while execution.	It is time sharing as the task assigned switches regularly.	The tasks are always further divided into sub tasks.	It carries multiple processors to execute the task.
4.	The processing is slower, as a single job resides in the main memory while execution.	Multitasking follows the concept of context switching.	It allows a single process to get multiple code segments.	A large amount of work can be done in a short period of time.

How to create thread in java?

We can create thread in java using two ways:

1). extends Thread: // Refer to Multithreading/extendThreadUse

By extending thread class, we can run thread in java by using thread class which provides the constructors and methods for creating and performing operations on the thread which extend the thread class that can implement Runnable interface.

We can use foll. constructors:

1. Thread();

2. Thread(Runnable r);

3. Thread(String thread_name);

4. Thread(Runnable r,String thread_name);

2). implements Runnable: // Refer to Multithreading/implementRunnable