C#.NET Multithreaded Programming FAQ

1. **What is Multi-tasking in c#.Net?**

   Its a feature of modern operating systems with which we can run multiple programs at same time example Word,Excel etc.

2. **What is Multi-threading in c#.Net?**

   Multi-threading forms subset of Multi-tasking instead of having to switch between programs this feature switches between different parts of the same program.Example you are writing in word and at the same time word is doing a spell check in background.

3. **What is a Thread ?**

   A thread is the basic unit to which the operating system allocates processor time.

4. **Can we have multiple threads in one App domain ?**

   One or more threads run in an AppDomain. An AppDomain is a runtime representation of a logical process within a physical process.Each AppDomain is started with a single thread, but can create additional threads from any of its threads. Note :- All threading classes are defined in System.Threading namespace.

5. **Which namespace has threading in c#.Net?**

   Systems.Threading has all the classes related to implement threading.Any .NET application who wants to implement threading has to import this namespace. Note :- .NET program always has atleast two threads running one the main program and second the garbage collector.

6. **How can we change priority and what the levels of priority are provided by .NET ?**

   Thread Priority can be changed by using Threadname.Priority = ThreadPriority.Highest.In the sample provided look out for code where the second thread is ran with a high priority.

   Following are different levels of Priority provided by .NET :-

   v ThreadPriority.Highest
   v ThreadPriority.AboveNormal
   v ThreadPriority.Normal
   v ThreadPriority.BelowNormal
   v ThreadPriority.Lowest

7. **How can you reference current thread of the method ?**

   "Thread.CurrentThread" refers to the current thread running in the method."CurrentThread" is a public static property.

8. **What's Thread.Sleep() in threading ?**

   Thread's execution can be paused by calling the Thread.Sleep method. This method takes an integer value that determines how long the thread should sleep. Example Thread.CurrentThread.Sleep(2000).

9. **How can we make a thread sleep for infinite period ?**

   You can also place a thread into the sleep state for an indeterminate amount of time by callingThread.Sleep(System.Threading.Timeout.Infinite). To interrupt this sleep you can call the Thread.Interrupt method.

10. **What is Suspend and Resume in Threading ?**

    It is Similar to Sleep and Interrupt. Suspend allows you to block a thread until another thread calls Thread.Resume. The difference between Sleep and Suspend is that the latter does not immediately place a thread in the wait state.The thread does not suspend until the .NET runtime determines that it is in a safe place to suspend it.Sleep will immediately place a thread in a wait state. Note :- In threading interviews most people get confused with Sleep and Suspend.They look very similar.

11. **What the way to stop a long running thread in c#.Net?**

    Thread.Abort() stops the thread execution at that moment itself.

12. **What's Thread.Join() in threading ?**

    There are two versions of Thread.Join :-

    v Thread.join().

    v Thread.join(Integer) this returns a boolean value.

    The Thread.Join method is useful for determining if a thread has completed before starting another task. The Join method waits a specified amount of time for a thread to end. If the thread ends before the time-out, Join returns True; otherwise it returns False.Once you call Join the calling procedure stops and waits for the thread to signal that it is done. Example you have "Thread1" and "Thread2" and while executing 'Thread1" you call "Thread2.Join()".So "Thread1" will wait until "Thread2" has completed its execution and the again invoke "Thread1".Thread.Join(Integer) ensures that threads do not wait for a long time.If it exceeds a specific time which is provided in integer the waiting thread will start.

13. **What are Daemon thread's and how can a thread be created as Daemon?**

    Daemon thread's run in background and stop automatically when nothing is running program.Example of a Daemon thread is "Garbage collector".Garbage collector runs until some .NET code is running or else its idle. You can make a thread Daemon by Thread.Isbackground=true.

14. **When working with shared data in threading how do you implement synchronization ?**

    There are a somethings you need to be careful with when using threads. If two threads (e.g. the main and any worker threads) try to access the same variable at the same time, you'll have a problem.This can be very difficult to debug because they may not always do it at exactly the same time. To avoid the problem, you can lock a variable before accessing it. However, if two threads lock the same variable at the same time, you'll have a deadlock problem. SyncLock x 'Do something with x End SyncLock.

15. **Can we use events with threading in c#.Net ?**

    Yes you can use events with threads , this is one of the technique to synchronize one thread with other.

16. **How can we know a state of a thread in c#.Net?**

    "ThreadState" property can be used to get detail of a thread.Thread can have one or combination of status.System.Threading.Threadstate enumeration has all the values to detect a state of thread.Some sample states are Isrunning,IsAlive,suspended etc.

17. **What is use of Interlocked class ?**

    Interlocked class provides methods by which you can achieve following functionalities :-
    v increment Values.
    v Decrement values.
    v Exchange values between variables.
    v Compare values from any thread. in a synchronization mode.
    Example :- System.Threading.Interlocked.Increment(IntA)

18. **what is a monitor object?**

    Monitor objects are used to ensure that a block of code runs without being interrupted by code running on other threads. In other words, code in other threads cannot

run until code in the synchronized code block has finished.SyncLock and End SyncLock statements are provided in order to simplify access to monitor object.

19. **what is ManualResetEvent and AutoResetEvent ?**

Threads that call one of the wait methods of a synchronization event must wait until another thread signals the event by calling the Set method. There are two synchronization event classes. Threads set the status of ManualResetEvent instances to signaled using the Set method. Threads set the status of ManualResetEvent instances to nonsignaled using the Reset method or when control returns to a waiting WaitOne call. Instances of the AutoResetEvent class can also be set to signaled using Set, but they automatically return to nonsignaled as soon as a waiting thread is notified that the event became signaled.

20. **what are wait handles ?**

Twist :- What is a mutex object ? Wait handles sends signals of a thread status from one thread to other thread.There are three kind of wait modes :-

v WaitOne.
v WaitAny.
v WaitAll.

When a thread wants to release a Wait handle it can call Set method.You can use Mutex (mutually exclusive) objects to avail for the following modes.Mutex objects are synchronization objects that can only be owned by a single thread at a time.Threads request ownership of the mutex object when they require exclusive access to a resource. Because only one thread can own a mutex object at any time, other threads must wait for ownership of a mutex object before using the resource. The WaitOne method causes a calling thread to wait for ownership of a mutex object. If a thread terminates normally while owning a mutex object, the state of the mutex object is set to signaled and the next waiting thread gets ownership

21. **What is ReaderWriter Locks ?**

You may want to lock a resource only when data is being written and permit multiple clients to simultaneously read data when data is not being updated. The ReaderWriterLock class enforces exclusive access to a resource while a thread is modifying the resource, but it allows nonexclusive access when reading the resource. ReaderWriter locks are a useful alternative to exclusive locks that cause other threads to wait, even when those threads do not need to update data.

22. **How can you avoid deadlock in threading in c#.Net ?**

A good and careful planning can avoid deadlocks.There so many ways microsoft has provided by which you can reduce deadlocks example Monitor ,Interlocked classes , Wait handles, Event raising from one thread to other thread ,ThreadState property which you can poll and act accordingly etc.

23. **What's difference between thread and process?**

A thread is a path of execution that run on CPU, a process is a collection of threads that share the same virtual memory. A process has at least one thread of execution, and a thread always run in a process context. Note:- Its difficult to cover threading interview question in this small chapter.These questions can take only to a basic level.If you are attending interviews where people are looking for threading specialist , try to get more deep in to synchronization issues as that's the important point they will stress.

24. **What are the two ways to create the thread?**

1.by implementing Runnable
2.by extending Thread

25. **What is the signature of the constructor of a thread class?**

Thread(Runnable threadob,String threadName)

26. **What are all the methods available in the Runnable Interface?**

run()

27. **What are all the methods available in the Thread class?**

1.isAlive()
2.join()
3.resume()
4.suspend()
5.stop()
6.start()

7.sleep()
8.destroy(

28. **What are all the methods used for Inter Thread communication and what is the class in which these methods are defined?**

1. wait(),notify() & notifyall()
2. Object class

29. **What is the mechanism defind by java for the Resources to be used by only one Thread at a time?**

Synchronisation

30. **Which priority Thread can prompt the lower primary Thread?**

Higher Priority

31. **How many threads at a time can access a monitor?**

one

32. **Which method waits for the thread to die ?**

join() method

33. **What are the two types of multitasking?**

1.process-based
2.Thread-based

34. **What is the signature of the constructor of a thread class?**

Thread(Runnable threadob,String threadName)

35. **What is the data type for the method isAlive() and this method is available in which class?**

boolean, Thread

36. **What is the procedure to own the moniter by many threads?**

not possible

37. **What is the unit for 1000 in the below statement?**
ob.sleep(1000)

long milliseconds

38. **What is the data type for the parameter of the sleep() method?**

long

39. **What are all the values for the following level?**

max-priority
min-priority
normal-priority
Ans : 10,1,5

40. **What is the method available for setting the priority?**

setPriority()

41. **What is the default thread at the time of starting the program?**

main thread

42. **How to refer to the current thread of a method in c#.NET?**

In order to refer to the current thread in .NET, the Thread.CurrentThread method can be used. It is a public static property.

43. **What are all the four states associated in the thread?**

1. new
2. runnable
3. blocked
4. dead

44. **Explain the purpose of Mutex class in c#.Net?**

System uses synchronization mechanism to make sure that a resource is used by only one thread at a time when more than one thread needs to access a shared resource at the same time.Mutex grants exclusive access to the shared resource to a single thread.A second thread that needs to acquire a particular mutex is suspended until the first thread releases the mutex.

45. **How does multi-threding work in c#.net?**

.NET has been designed from the start to support multi-threaded operation. There are two main ways of multi-threading which .NET encourages: starting your own threads with ThreadStart delegates, and using the ThreadPool class either directly (using ThreadPool.QueueUserWorkItem) or indirectly using asynchronous methods (such as Stream.BeginRead, or calling BeginInvoke on any delegate).
In general, you should create a new thread "manually" for long-running tasks, and use the thread pool only for brief jobs. The thread pool can only run so many jobs at once, and some framework classes use it internally, so you don't want to block it with a lot of tasks which need to block for other things. The examples in this article mostly use manual thread creation. On the other hand, for short-running tasks, particularly those created often, the thread pool is an excellent choice.

46. **When two threads are waiting on each other and can't proceed the program is said to be in a deadlock?**

True/False

True

47. **How to pause the execution of a thread in c# .NET?**

The thread execution can be paused by invoking the Thread.Sleep(IntegerValue) method where IntegerValue is an integer that determines the milliseconds time frame for which the thread in context has to sleep.

48. **Garbage collector thread belongs to which priority?**

low-priority

49. **What is meant by time slicing or time sharing?**

Time slicing is the method of allocating CPU time to individual threads in a priority schedule.

```
public void Func()
{
for (int i = 0; i <5; i++)
{
Globalvar += 1;

}
}
```

50.
```
void main()
{
Thread x = new Thread(Func);
Thread y = new Thread(Func);

x.Start();
x.Join();
y.Start();

Write(Globalvar);
}
```

What is the output?

The question was about thread concurrency.Two threads run the same code, that increments global variable 10 times each.If both threads enter the critical section at the same time then final result would be 10 (instead of 20), but it can be even less than 10.