



Transcript of the Web Chat on  
**Multi-threading Lifecycle  
& Threading Tools**



Q: How do the multiple processors connected in a system communicate together? Is there something like master and slave or anything else?

A: Through FSB, looking at a snoop filter which helps allocate threads based on the availability of the processor or core.

Q: What is FSB?

A: front side bus which is basically used by CPU to communicate with the memory

Q: How does memory management work in multi processor environment? How do all the calls to memory get synchronized?

A: In a multi-processor environment, memory synchronization is a responsibility of the user. The OS and the hardware don't assure this synchronization. So, inherently, all the memory allocation and management routines are serial in the OS. There are some memory allocation and management software packages like SmartHeap library which allow seamless synchronization and allocation of memory on multi-processor and multi-thread environment

Q: What are the common reasons that fail the program to execute on processor?

A: If the program itself takes too much memory that clogs the system memory

Q: What are the common reasons that make the program fail to execute on the processor?

A: A program may fail if not enough stack is allocated or the environment that is given to the processor to function in terms of memory, or if we have logical errors in a program

Q: Is there any chance of Deadlock being caused between processors?

A: Deadlock occurs between threads even if these are on two different processors. So, yes, deadlock occurs between processors each of which is executing threads which are sharing the same code/ data or synchronization objects.

Q: How do I run my thread on a specific processor?

A: You can set thread affinity. There are APIs in Windows and Linux that will help set affinity to a specific processor core. Also if you are familiar with OpenMP, it has some APIs that allow you to set thread affinity.

Q: How do multiple processors work together in multithreading?

A: When there are multiple processors in a machine, the Operating system detects the number of processors in the machine and it schedules the threads to the CPU based on that. This way the OS effectively utilizes the processors optimally. I hope this answers your question. Let me know if you need more clarity.

Q: Thread is light weight process share same resource, Can you please explain that?

A: Yes, you can think of a thread to be a light weight process. However, please note that thread and process have different privilege levels, access to memory, and visibility in OS.

Q: Can we create a thread in C language, if yes how?

A: Yes, you can create thread in C language using POSIX thread mechanism.

Q: Monitor and lock are same mechanism or different with respect to java and OS, explain separately

A: Monitor and lock are way different on an OS versus Java. The implementation works this way on Java. There are primitives that are used for synchronization between threads. With respect to the OS,

it is a physical lock on the hardware/CPU. The OS directly controls the hardware, whereas in Java it is more of a software implementation.

Q: How we can achieve better performance when we are executing two or more thread. How to schedule /set priority of thread?

A: There are various ways. 1) Load balance the work to your threads 2) Assign the correct number of threads, meaning don't underutilize or over utilize the available cores in your CPU. You can use functions available in the API to set the priority to your threads

Q: Which is the fastest processor of Intel?

A: On the desktop side it is the Intel® Core™ i7 Processor, but it depends on the usage model

Q: Which is smallest process of Intel?

A: Intel® Atom™ Processor is smallest processor built with the world's smallest transistors. This microprocessor powers a new category of devices called Mobile Internet Devices (MIDs), and another new category of devices for the internet called netbooks and nettops.

For more look at <http://www.intel.com/technology/atom/>. To know more about MIDs built on Intel® Atom™ Processor, look at <http://www.intel.com/products/mid/> To know more about Netbooks, <http://www.intel.com/Consumer/Learn/Internet-Devices/atom-netbook-detail.htm>

Q: How to restrict CPU scheduler to select a certain thread?

A: Using thread affinity, you can restrict the thread to run on a specific processor.

Q: Which is better? To make a thread and extend a Thread class or implement Runnable interface in not book manner

A: Unless you are interested to overload and override Thread class member functions, it would be better if you use the Runnable interface.

Q: What is difference between wait ,halt and sleep with respect to java

A: Thread.Sleep - Sleeps a thread for a moment

Thread.wait - Thread waits unless notified

Thread.halt-Execution of that thread without affecting other threads

Q: We create a thread object and call start() method ,again we call start() method on same instance is not possible in java. Is it possible to again start thread other language?

A: You cannot call start on the same instance in any language

Q: How does JVM map thread with OS native threads explain

A: To achieve maximum performance on a operating system, JVM uses the native threading platform of the operating system. Windows and Linux Operating Systems use 1:1 mapping that means every 1 user thread has a 1 kernel thread assigned.

Q: I want to know how one thread communicates with other thread

A: There are many mechanisms to communicate between threads, like wait, waitforall etc. Also Intel® Threading Building Blocks (TBB) has got many mechanisms to communicate with each other. Please visit our web page and download Intel® TBB. <http://www3.intel.com/cd/software/products/asmo-na/eng/index.htm>

Q: Is there any mechanism by which we can implement JVM specific thread

A: I am not aware of any mechanism to implement JVM specific thread

Q: Which is the best scheduling technique to select a process and / or thread

A: The best scheduling mechanism is a 1:1 mapping between the thread and CPU core. Now the basic unit of work is a thread and not a process. All the latest OS are aware of this and the OS will take care of the scheduling. So the user doesn't have to worry about the scheduling

Q: Which language gives optimal performance to create and execute a thread

A: Languages like C, C++ and Fortran which get executed natively on the CPU are the best for threaded programming. In case Java or .NET they don't compile to the native code(because they compile to Intermediate Language) and hence the performance might not be as great as the one with C or C++ or Fortran. Here the .NET CLR and JVM are optimized to take advantage of threading. I hope this answers your question.

Q: What is synchronization with respect to a thread?

A: Thread synchronization refers to the act of shielding against multithreading issues such as data races, deadlocks etc.

Q: Please tell me about join() method with respect to java thread

A: join() method is used to join the threads that were launched in an application. This method is going to take the thread id and joins that thread to the main thread.

Q: Is there any free software tool which analysis working of thread with respect to java?

A: I am not aware of any JAVA tools for analyzing threads available free.

Q: Which is the smallest processor of Intel?

A: Intel® Atom™ is smallest processor built with the world's smallest transistors. This microprocessor powers a new category of devices called Mobile Internet Devices (MIDs), and another new category of devices for the internet called netbooks and nettops. For more look at <http://www.intel.com/technology/atom/>

Q: Is there a capacity for thread pool in which the thread resides?

A: Can you please be specific? You can create as many threads in a pool as you need. This will depend on the support of the APIs. Please check for specifics.

Q: When we build a processor, what are the most important things? Please explain hierarchy wise.

A: 3 P's: performance, power and price

Q: Will there be a dead lock problem when we synchronize two block codes with two instances?

A: Then your application should work fine. Please make sure, there is no deadlock in your application using Intel® Thread Checker. Please visit, <http://www3.intel.com/cd/software/products/asmo-na/eng/index.htm>

Q: What is daemon thread & how it differ form user thread?

A: Daemon thread is a normal software thread created by the application which runs in the background (meaning that it is not tied to any user interface devices - keyboard, mouse or monitor). The only way to kill the daemon process is by signaling mechanism supported by the operating system.

Q: I read that thread share same resource through context switching. How is it possible?

A: I think if you want to share resources, then it is best to use threads because it is easy to context switch and share resources without much performance loss as compared to using processes instead.

Q: Which scheduling technique is used by Windows OS to select a thread?

A: Preemptive scheduling with timeslice.

Q: How many threads a process can have? Is it OS specific and also... Is it Processor specific??

A: There is no limit in most of the OS. It just depends on how much memory there to hold these threads and execute them

Q: How many threads can a process have? Is it OS specific? Do we have a limitation on the basis of the processor being used?

A: This typically depends on the OS. Inherently on Microsoft Windows OS, there isn't any limit on the number of threads per process. This depends on your available RAM and the stack size that you allocate per thread. There isn't any limitation on the processor front. For Linux, we have this limitation that can be set using ulimit command.

Q: Do all multi-core CPUs support 64 bit software?

A: Most of the recently launched Intel multi-core processors support 64 bit.

Q: Can you please describe the Basic things about multithreading

A: Multi-threading basically means programming to use more than 1 thread to execute a given task. This can be achieved by using thread APIs available in several programming languages. Multithreading programming helps in getting better performance when using multi-core processors for execution of a given problem or a task.

Q: How to Monitor help in thread synchronization?

A: Monitor class has got bunch of API's that help multithreading. This will be used for synchronization of threads and managing all the threads that were created. I assume that your question was based on .NET monitors. If not please revert back so that we can answer your question.

Q: Can we manage more than thread in STA?

A: Single Thread Apartment which means the Windows\* controls will be running in thread that will be started on initialization. When another thread tries to access the controls then it is not possible. Cross thread communication is not supported. Delegates has to be used.

Q: Can we manage threads in STA arceture?

A: The Single-Threaded Apartment (STA) model is very commonly used. Here, a COM object stands in a position similar to a desktop application's user interface. In an STA model, a single thread is dedicated to drive an object's methods, i.e. a single thread is always used to execute the methods of the object. In such an arrangement, method calls from threads outside of the apartment are serialized and automatically queued by the system (via a standard Windows message queue). Thus, there is no worry about race conditions or lack of synchronicity because each method call of an object is always executed to completion before another is invoked.

Q: What is Wait-one and Wait all in Threading?

A: refer to <http://msdn.microsoft.com/en-us/library/kzy257t0.aspx> for more. They are Microsoft APIs

Q: Is Multithreading different for different OSs?

A: Both Linux\* and Windows\* are threaded, there is no major difference in the way the threading is done at OS level

Q: What is the basic requirement to use Thread in .Net?

A: First you need to decide if your application needs the benefits of using threads such as performance/speed, responsive UI to the user, etc. With this clarity, you can design your application for threads. .NET has been designed from the start to support multi-threaded operations. 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).

Requirments for introducing Threads in your application. Why Use More Than One Thread?  
Many single-threaded client applications already exist, and many more are being written daily. In many cases, single-threaded behavior is sufficient. However, for certain applications, adding a little asynchronous behavior can improve the user's experience.

Q: What is the basis of Threading in WIN XP\* and WIN 2003 Server\*?

A: Please read the article [http://msdn.microsoft.com/en-us/library/ms684841\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms684841(VS.85).aspx) for more information.

Q: The above mentioned Threading tools are helpful to write the synchronous program or check the Mutex object in API?

A: Yes. Intel® Threading Building Blocks is helpful to do that.

Q: I have a 2.21 Ghz Core 2 Duo E4500 Processor and I want to maximize the performance and the speed of my processor, is there any way I can improve the performance. What settings can be modified for that?

A: There are some BIOS settings that you can use to improve the performance of your application. Note that the performance of your application is measured based on whether it is utilizing full CPU and so on.

Q: Please tell me the settings or methods to get optimum speed and performance out of the processor

A: It depends on the processor you have. Based on the processor, settings are subject to change. In order to optimally utilize the current processors, you have to ensure that the software is multi-threaded and all the cores of the CPUs are being utilized. If you have any specific questions on the settings, please do let me know.

Q: How to optimize this?

A: Here are my recommendations. Ensure that all of the cores are enabled in the BIOS (two cores). Also the application you use should be multithreaded in order to effectively utilize the two cores of the CPU. Ensure that EIST (Enhanced Speed step technology) is enabled in the BIOS. Also ensure that your OS Power scheme is set to the performance mode (if you are using Vista). Let me know if you need more tips specifically in optimizing any application you develop.

Q: What is race condition in multithreading?

A: When multiple threads try to write at the same memory location or when one thread is reading and the other is writing, it's called race condition. We should use synchronization variables to avoid race conditions.

Q: What is hyper threading?

A: You can take a look at <http://en.wikipedia.org/wiki/Hyper-threading>

Q: What is the difference between starvation and deadlock?

A: Deadlock is if the thread set is not able to progress further, as they cannot release locks

Q: Is the JDBC-ODBC Bridge multi-threaded?

A: No. The JDBC-ODBC Bridge does not support concurrent access from different threads. The JDBC-ODBC Bridge uses synchronized methods to serialize all of the calls that it makes to ODBC. Multi-threaded Java programs may use the Bridge, but they won't get the advantages of multi-threading.

Q: How do you suspend and resume threads?

A: in POSIX API (commonly in Linux/Unix platforms), `pthread_suspend` and `pthread_continue` api calls. In .net, `Thread.Suspend` and `Thread.Resume`, In Win32, `SuspendThread` and `ResumeThread` API.

Q: What are all the methods used for inter communication?

A: Could be Inter Process, Inter Thread or shared memory communication

Q: What is simultaneous multithreading?

A: Simultaneous multithreading is hyper threading being reintroduced on the new Intel® Core™ i7 platform, with this feature the OS is tricked into believing that there are two CPU's per core therefore allowing the platform to support twice the number of software threads with half the number of cores

Q: Can a thread become an orphan thread like an orphan process?

A: A zombie process is one which has died, but its parent hasn't wait()'d on it to read its exit status, so it's still tying up system resources... Similarly, you can have a situation where a thread has exited, but none of the other threads has done a `pthread_join()` on it to get its exit value, so it's also tying up resources... You can avoid that by doing `pthread_detach()` on any threads you don't care about the exit values of...

Q: Name the superfast Youtube downloader with resume download?

A: CyberLink PowerDirector has that feature

Q: What is hyper threading?

A: <http://en.wikipedia.org/wiki/Hyper-threading>

Q: What is the difference between starvation and deadlock?

A: when one thread consumes all the CPU cycles, and does not allow other thread(s) to execute, then its starvation

Q: Can you explain Thread safety?

A: Thread safety is a computer programming concept applicable in the context of multi-threaded programs. A piece of code is thread-safe if it functions correctly during simultaneous execution by multiple threads. In particular, it must satisfy the need for multiple threads to access the same shared data, and the need for a shared piece of data to be accessed by only one thread at any given time.

Q: If at all any given thread monopolizes CPU time?

A: Yes you can. It is based on the thread Priority. If one of the threads is completely occupying your CPU, you can increase the priority of other thread and deprioritize the existing thread that's occupying the CPU currently.

Q: What is HashMap and Map?

A: Map is Interface and Hashmap is class that implements that.

Q: What is the difference between time slicing and time sharing?

A: Time slice is very fixed, where as time sharing is not fixed. It's dynamic. For eg if the time slice for a thread is 10 mins, it is fixed and can't change. Whereas in time sharing, if its 10 mins, multiple threads can share between the specific time

Q: What is Lockfree?

A: Lockfree is a non blocking implementation of any algorithm without any synchronization primitives might affect the scalability of the multi threaded application

Q: Why should we not use TerminateThread-esque functions?

A: Can you give more details on esque functions? You can use terminate thread functions, but it might not be safe to do so especially when the thread holds resources which aren't released or if other threads are waiting for the resources. As you might end up in deadlocks or race conditions.

Q: What is cache line?

A: Cacheline is a collection of values that are loaded together into the cache when a load instruction for a value is executed. Cacheline is used to maintain temporal locality so that contiguous values are loaded and available for usage next.

Q: Provide two programming examples of multi-threading giving improved performance over a single threaded solution?

A: The Intel® Software Network has many resources explained in detail. Please visit [www.intel.com/software](http://www.intel.com/software)

Q: What is the difference between hyper-threading, multi-threading and super-threading in computer processors?

A: Never heard of super threading :), Hyper threading technology, some of the resources are shared and some are duplicated so give you that additional performance boost. In server applications (well threaded applications) you might see up to 30% performance improvement and in some cases you might not. OS sees this as separate core (if you look in task manager) and schedules threads.

Q: What is a core 2 quad computer?

A: Intel® Core™2 Quad processor is a CPU with 4 cores on it

Q: What is the difference between pthread and thread?

A: Pthread is a threading library for implementing threads.

Q: How will time be allocated for each thread?

A: Time will be allocated to threads based on the priority you set. If the priority is normal, then it will be executed on a round robin fashion. If priority is set, it would ensure that the thread with a higher priority is executed first.

Q: What is MESI?

A: MESI is a protocol for cache coherency. M for Modified, E- Exclusive, S- Shared, I- Invalid. These are the states that a cacheline would be in.

Q: What is Multi Threaded Server MTA ?

A: If the COM object's methods perform their own synchronization, multiple threads dedicated to calling methods on the COM object are permitted. This is termed the Multiple Threaded Apartment (MTA). Calls to an MTA object from a thread in an STA are also marshaled. A process can consist of multiple COM objects, some of which may use STA and others of which may use MTA. The Thread Neutral Apartment allows different threads, none of which is necessarily dedicated to calling methods



on the object, to make such calls. The only provision is that all methods on the object must be serially re-entrant

Q: What is zombie process?

A: Zombie process is a process which is in a quasi state - that is it is neither dead nor alive and doesn't respond. Typically this happens when you have inefficient join or creation or termination threads.

Q: What is chip multithreading and chip multi processing?

A: Chip Multiprocessing (CMP), duplicates or multiplies the entire processor core with almost all of its subsystems on a single die. It is also possible to "simply" co-package two already existing and only slightly modified processors with some additional logic, which will behave exactly as a dual-core die. Chip multi-threading (CMT) brings to hardware the concept of multi-threading, similar to software multi-threading. Software multi-threading refers to the execution of multiple tasks within a process. The tasks are executed using software threads. The software threads are executed on a single processor or on many processors simultaneously.

Q: Why do I get the error "org.eclipse.swt.SWTException: Invalid thread access"?

A: This is outside the scope of this forum. It would be useful if you could raise this issue on the eclipse forums for a suitable answer

Q: Difference between TCP/IP and OSI layer?

A: TCP is a Transport layer protocol and IP is a network layer protocol. For more information, visit wikipedia @ [http://en.wikipedia.org/wiki/OSI\\_layer](http://en.wikipedia.org/wiki/OSI_layer)

Q: Have we to use single-threaded or multi-threaded servlets?

A: Servlets are normally multi-threaded. The servlet container allocates a thread for each new request for a single servlet without any special programming. Each request thread for your servlet runs as if a single user were accessing it alone, but you can use static variables to store and present information that is common to all threads.

Q: How do you set thread priorities?

A: Setting thread priorities can be done using APIs. Windows\* and Linux\* have separate APIs that is used for controlling the thread priorities. Let me know if you are interested in knowing the specific APIs.

Q: What is the priority of the garbage collector thread?

A: The priority assigned to GC threads is very high. It really depends on which GC collector is being used at runtime

Q: What is deadlock and what is live lock?

A: Deadlock is performance limiting situation. It happens dynamically. It happens when threads contest to have the same piece of data. Live lock is a state where two threads continuously change their state in response to changes in the other, resulting in both of them not progressing to complete.

Q: What is cache coherency?

A: Cache coherency is used to ensure that the cache values are all coherent across processors and caches.

Q: How to make threads share CPU almost equally?

A: To make sure your CPU share equally of any workload, you have to use Intel® Thread Profiler. This tool helps you to find out how much work you are allocating to Core-1 OR Core -2

Please Visit <http://www3.intel.com/cd/software/products/asmo-na/eng/index.htm>

Q: I just want to know what is a Thread Profiler?

A: Intel® Thread Profiler is an Intel tool that will provide detailed diagnostics on the performance of multithreaded applications in terms of load balancing, granularity and synchronization.

Q: How to use Circular Buffer/

A: Implementation of a double linked list where the left pointer in the head, point to the end node and the right pointer points to the last node. In essence, you would be able to circulate through the entire list.

Q: How to perform synchronous and asynchronous task in multithreading lifecycle?

A: The support of launching synchronous and asynchronous tasks depends on the APIs supported on the OS and the programming language. Typically the OS and the languages support launch of asynchronous and synchronous tasks for multi-threading effectively. Please refer to the OS and language API for details.

Q: If there are 2 thread which call 2 different Synchronized methods using an object...What will happen

A: Since you have 2 threads they call 2 different synchronized methods they run in parallel if each thread has instance of that object. If the object through which you are calling needs to be atomic then you take a lock on that object so only 1 guy can call the function.

Q: If some x thread is stopped abruptly.....will the daemon thread clear the obj used in that x thread? If that is so then how do performance issues occur?

A: Unless there is an application logic for the daemon thread to handle this scenario, the daemon thread will not be able to handle it. If the stopped thread was performing some critical task then there will definitely be a performance hit.

Q: How does Synchronization help in Multithreading?

A: Synchronization helps in multithreading scenario where you need to provide per thread exclusive access to shared variables so that the overall correctness of the application can be maintained.

Q: If a thread is forced to stop abruptly there will be a performance issue, then when will the Demon thread be used?

A: Can you please clarify how the daemon thread and the current thread related

Q: How do thread pool works?

A: In a thread pool, a number of threads are created to perform a number of tasks, which are usually organized in a queue. Typically, there are many more tasks than threads. As soon as a thread completes its task, it will request the next task from the queue until all tasks have been completed. The thread can then terminate, or sleep until there are new tasks available.

Q: One function in .Net is consuming CPU up to 40 %. It is in a continuous loop for receiving messages. How can I minimize it?

A: You might want to first investigate where in the application the maximum time is spent and then investigate what code changes can be done to improve performance.

Q: Is c# is good for creating threads?

A: C# supports parallel execution of code through multithreading. A thread is an independent execution path, able to run simultaneously with other threads. .NET has been designed from the start to support multi-threaded operation.

Q: What exactly is a thread pool?

A: Thread pool is a pattern of programming, where a number of threads are created to perform a number of tasks, which are usually organized in a queue. Typically, there are many more tasks than threads. As soon as a thread completes its task, it will request the next task from the queue until all tasks have been completed.

Q: Is there is any need to kill a thread once created?

A: No, when the threads finish the work, and when the processor exits it gets terminated.

Q: What are disadvantages of sleep method?

A: Sleep method forces the thread to sleep for a specified duration of time but the processor on which the thread is getting executed is not released. If the thread has to sleep for a short period of time, then that might be useful to retain it but if the thread has to sleep for a longer time, it would be better if the thread voluntarily yields the processor.

Q: What is thread pool?

A: A thread pool is a collection of threads, which you keep "alive" and use/reuse to process incoming "tasks". When a new task arrives (a typical example is a request to an HTTP server) you try to find a thread from the collection, which is idle, and hand the task to it. If no such thread exists you either wait for one to become available or add a new thread to the pool (usually there is an upper limit). After the thread has finished processing the task, it is not terminated, only marked as idle and ready to be reused for another task.

The main advantages of using a thread pool as opposed to creating a new thread to handle each new task are:

- 1) By reusing threads you save the thread creation/destruction overhead.
- 2) You have control over the maximum number of tasks that are being processed in parallel (= number of threads in the pool).

Q: How can CPU usage by any application be minimized by threading?

A: Threading brings in parallelism into a software, this can be used to split the workload into multiple parallel threads which will then get distributed across cores in a multi-core platform, this will make sure that the overall CPU usage and frequency is lower

Q: How can find the completed history of the process?

A: You can use tools like perfmon to create a log of all the completed processes and that will help identify the list of completed processes.

Q: How can I handle the deadlock?

A: Deadlock is purely handled by synchronization. If all the threads are synchronized properly, deadlock might not happen.

Q: I want all the Q's and A's. Is a copy available?

A: All questions and answers will be archive on Intel® Software Network. Please visit <http://software.intel.com/en-us/articles/web-chat-series> for the chat transcript

Q: What is the max no of multithreads supported with out system performance dropping?

A: Typically we would want to create as many threads as the number of processors. If you create too

many threads, you would end up switching between threads which would result in performance issues.

Q: When I put my laptop in sleep mode. is there any effect to mother board?

A: There is no effect

Q: Are there any improvements in the multi threading concept from .Net 1.0 to .Net 3.5?

A: There are multiple improvements in the garbage collector and then multiple APIs have been introduced for writing efficient multi threading code and performance has increased in general for multithreading.

Q: What is thread safety?

A: A piece of code/function/api is thread-safe, if it functions correctly during simultaneous execution by multiple threads

Q: It will be helpful if the questions and answers are shared after the session is completed

A: The transcript of the chat will be archived and available to you at <http://software.intel.com/en-us/articles/web-chat-series> Please book mark this page for future reference.

Q: When we refer to .Net applications the threading concepts are used very rarely even though they are useful to increase performance. Why is that?

A: That's not true, many .Net application are threaded. Garbage Collector in .Net has been threaded.

Q: Whose responsibility is thread switching and based on what are the threads switched?

A: Thread switching is done by the OS. The threads are switched based on the time slice assigned to the threads by the OS and their priority.

Q: What exactly was a mobile processor concept and why it has been discontinued? What I mean is that now Intel has stopped using the term Mobile processor while promoting and advertising for new generation of Laptop processors

A: Can you clarify what you mean by Mobile Processor. Mobile processors have not been discontinued. They are very essential part of our roadmap. Refer to Intel® Centrino® on our web site [www.intel.com](http://www.intel.com)

Q: Does RAID improve Multithreading. Why I'm asking is that for DB activities with Oracle or DB\_2, the space is more concerned with the number of threads involved.

A: Databases are very compute, memory and IO Intensive applications. Many DBs utilize different techniques including multithreading to achieve maximum performance. RAID always helps IO Intensive applications. Using multithreading and RAID may help achieve better performance on a multi-core platform.

Q: Can you please provide simple C++ program to show an example on multi threading

A: Please visit Intel® Software Network at [www.intel.com/software](http://www.intel.com/software) You will be able to find the information you are looking for.

Q: I heard that Sony PS3 processor is capable of providing high end graphics with Wi-Fi ie. Multi-threaded. Does Intel have any plan to compete with this one, since Gaming Console Do need Multi Threading concepts!

A: Intel based PC platforms are also capable of supporting hi-end graphics. The current generation can support shader model 4, DX 10.5 and Open GL, which is the highest quality of graphics available. The latest Intel® Core i7™ platform is the best gaming platform available in the market.

Q: Is the Intel New Duo Core more capable than it's predecessors?

A: The new family of processors that are now multi-core, have better power optimization and new features such as SMT and QPI

Q: In a Load Balancing Web Application, do you suggest multithreading in all the servers or any one?

A: By default, most web servers are multithreaded. In a single platform running a single instance of web server, the server will be able to handle multiple requests. Whether it is a single platform environment or a load balancing environment, each server in the environment will be able handle multiple requests

Q: Is this Webinar focused on Intel Microprocessor Multi Threading concepts or the Software based Threading concepts

A: This webinar will answer both of these, you can ask us general questions about software threading and Intel microprocessor multithreading

Q: Can you please provide simple C++ program to show example on multi threading !

A: Please visit [www.intel.com/software](http://www.intel.com/software) and check out the Smoke Demo program

Q: Is there any way to do user defined multithreading?

A: Yes, user defined multithreading is implementable using the APIs available in the programming languages. These include typically the Windows Win32 API, Linux pthread API and Java and .NET API.

Q: Balaji: Which OS is better for Server? I mean OS with Multithreading for Optimized Performance

A: Most of the current modern operating systems have already been optimized for multithreading.

Q: Please explain the Life Cycle of Multi threading

A: Threading involves four major stages, Analyze, Code, Debug and Verify.

Intel has tools for every stage. Please visit <http://www3.intel.com/cd/software/products/asm-na/eng/index.htm> to access tools for multi threading.

Q: In case of older co-processor concepts or latest L2 cache one, which has got more predominance in multithreading

A: Can you explain the question little better. L2 helps if you have threads which have large working sets. Hence the performance boost is very good for threading if you have applications that carry large data sets. We have been shipping with good L2 cache and L3 cache.

Q: Like the Garbage collection in Java, Does Intel provide any automatic clearance of unused threads and for my earlier question I mean how to migrate from old concept based processor to L2/L3 levels without under clocking.

A: No, Intel does not provide any functionality for clearing unused threads. Can you please clarify the second question?

Q: Can you please explain Over Clocking with reference to Multi threading?

A: Over clocking is a concept in which the processor frequency is increased beyond the standard rating of the processor when it is manufactured or packaged. Multi-threading is a concept where there are multiple processors for execution in a single machine. Thus when writing multiple threads for doing a task, you get better performance on a multi-core machine. Over clocking gives benefit because the processor is running at a higher frequency or clock speed completing the task faster.

Q: Is there any way to run and start hardware threads like I do in Java

A: In x86 architecture, there is no direct correlation between the Java Threads and Hardware threads. The Operating System scheduler handles all the requests and will assign it to a processor based on the load.

Q: In the critical section region of the OS only one process executes, but can we have multiple threads executing?

A: No, critical section is protected by locks, so that the data is left consistent in read-write conflicts. So to do the work atomically, we will use critical section so only 1 thread executes the code inside that block.

Q: what is the difference between a thread being suspended and thread being blocked

A: Thread Blocked - A thread can enter in this state because of waiting for the resources that are held by another thread. Thread suspended state means the thread will be in wait or sleep mode.

Q: I have a question for the panelists. How can one ensure that one thread doesn't monopolize CPU time even when it is waiting for I/O

A: Generally the OS ensures all threads are given fair chance of execution. So no thread can monopolize and hold other threads from executing. Even when you set highest priority on a thread, the OS ensures that the lowest priority thread gets a chance to execute.

Q: Can you just please explain the multithreading life cycle in core dual processor of Intel?

A: Analyze, code, debug and verify are the 4 stages in the multithreading lifecycle. Intel has software tools to use at every stage, including compilers, performance analyzers, threaded libraries, thread checkers etc. Please refer to [www.intel.com/software](http://www.intel.com/software) to know more on these tools.

Q: When threads are abruptly stopped does it affect the efficiency of the process?

A: Yes, threads have their data structure, and stacks, and memory, which need to reallocate each time you stop, kill and start threads. That's the reason why we have the concept of a thread pool. If the thread is abruptly stopped, it may lead to fatal error, or stack or heap corruption because the exception handling is not done properly.

Q: When dual processors end in a deadlock then how do we recover from deadlock?

A: You can avoid deadlocks using any synchronization mechanisms like mutexes, locks etc.

Q: How efficient does multithreading prove in the area of multimedia?

A: Multithreading can help multimedia performance in many ways. Multithreading can be leveraged in many formats. For e.g. when you are decoding a large image, you can use threading to decode different parts of the image using threads. In MPEG processing, you can assign different I Frames to different threads. You can also use threads to handle IO and processing separately. Hence the answer is - threading can be very useful to get most out of CPU especially for Multimedia applications. Multimedia applications stress more parts of systems and hence thread can do a better job.

Q: Deadlock occurs when all the 4 issues hold

A: can you please clarify the question

Q: What are the other languages multi-threading programs can be written in other than Java?

A: C, C++, Java - all the major languages support multithreading

Q: How does the traditional languages like C, C++, Java multithreading concept differ from that of LISP

A: LISP is pure functional programming language which assumes that there is not mutable state and all the state resides within the function itself where C/C++/Java are imperative languages that describe computation in terms of statements that change a program state. Multithreading concepts will remain common to both these programming language concepts

Q: Can you let me know about multi-core dual processor?

A: Dual processor means, two different processors on the same machine. Each of this processor can again be a single core or dual core or even a multicore processor. If a machine has got two processors and each of them are dual core processors, then essentially the machine can handle  $2 \times 2 = 4$  threads in parallel. I hope this answers your question. Let me know if you need more clarity.

Q: What are the other multithreading tools available, other than the Intel ones?

A: As I know Only Intel® Threading tools are very useful to parallelize your application, let me know if you have any questions about Intel® Tools. Sorry, I am not sure about non Intel tools for threading.

Q: Can you give the threading tools which are specifically for Java?

A: Intel® VTune™ can be used to profile your application. Intel® Performance Libraries (IPP and MKL) can also be used from Java to JNI

Q: What is the maximum amount of threads that won't compromise on efficiency of the system

A: The answer depends on the amount of Compute resources in the system. For example, if there are two cores, then ideally only two threads can run at any time. However if you have lot of memory and can hold large working set (Process size) then you can span multiple threads.

You need to find a balance. Much depends on the application architecture. Sometimes more threads mean less performance because of constant spooling.

Q: How many cores does Atom processor have?

A: Intel® Atom™ processor has one core but can support 2 threads thanks to the hyper threading feature on the proc

Q: What is thread switching?

A: Thread switching is same as Context Switching.

Q: What are the assembly languages that you are using in your processor?

A: The assembly language used in our processor is the Intel64 assembly language.

Q: what is the performance of Intel new processors with Windows platform

A: Please go to <http://www.intel.com> which has some benchmark numbers that you can look at.

Q: Is context switching the same as threads?

A: Context Switching happens when one thread has completed its task and the other thread has to execute. The status of the existing thread has to be saved and switched to the other thread, which is called Context Switching. More details can be found in Wiki:

[http://en.wikipedia.org/wiki/Context\\_switch](http://en.wikipedia.org/wiki/Context_switch)

Q: How synchronization work?

A: Synchronization is a mechanism to achieve mutual exclusion for individual thread access to shared variables. In first thread would have to take a lock on a shared variable, updates the variable and then

releases the lock. In the second thread, if it tries to take a lock, the thread will have to be blocked due to mutual exclusion behavior. Only when the first thread releases the lock, the second thread will acquire the lock and continue to update the critical section and then release the lock once the update is complete.

Q: What is multithreading models?

A: Multi-threading models define how a given problem is solved using multiple threads. These include functional decomposition, domain (data) decomposition, and pipeline decomposition. In addition, multi-threading programming guidelines define the way a program is to be split and communication patterns to implement threading effectively.

Q: What is the difference between Core2duo and Dual core processor

A: Dual core processor is a category. It refers to all processors with 2 cores. Intel® Core™2 Duo is an Intel brand that refers to a dual core processor

Q: Can you explain IPC?

A: IPC is Inter process communication or instructions per clock tick. Please let us know which IPC you are referring to?

Q: How is the internal and external cache system organized in core 2 duo and quad core processors?

A: Can you please clarify on the external cache system

Q: How does .Net work on Intel processors?

A: The computing industry (including Intel processors) is trending towards multi-core processors. .Net supports multithreading both on client applications and web based applications (thread pools, ASP, .NET threading model, asych & call back design mechanisms).

Q: What is the technology behind new processor Intel Core i7?

A: The Intel® Core™ i7 platform is based on 32 NM Nehalem architecture. This brings in a new memory implementation known as quick path interconnect and also brings in support for simultaneous multithreading. It also has turbo boost which allows the CPU to automatically over clock

Q: Can you tell me about the technology behind Intel Core 2 Duo?

A: The current family is based on Penryn and core micro architecture which supports multithreading and better HD features.

Q: Can you explain more about forthcoming Intel processors?

A: Intel® Core™ i7 was just launched. This year there will be mainstream DC and QC desktop and laptop processors which will support hyper threading.

Q: What is the performance of .Net on Intel processors?

A: We are closely working with Microsoft to get better performance of .Net applications. As I know, you should get good performance out of any .Net applications on Intel platforms.

Q: can you explain about Intel dual core processor and its cache system

A: You can visit <http://www.intel.com/software> for some articles and you can also look @ manuals to understand it more. Manuals are available @ <http://www.intel.com>

Q: What are the advantages of new processor compared to existing one?

A: Better features, more power efficiency and support for a larger number of software threads



Q: During the sleep method what is happening inside the processor?

A: When sleep method is executed, the processor is executing halt (HLT) instructions.

Q: What is IPC?

A: Inter-Process Communication (IPC) is a set of techniques for the exchange of data among multiple threads in one or more processes. Processes may be running on one or more computers connected by a network. IPC techniques are divided into methods for message passing, synchronization, shared memory, and remote procedure calls (RPC). The method of IPC used may vary based on the bandwidth and latency of communication between the threads, and the type of data being communicated.

Q: What are differences (if any) between CPU threads and GPU threads?

A: The concept of a thread is the same whether it is a GPU or CPU. The way in which it is scheduled is different on a CPU and a GPU. The OS handles it little differently. CPU threads can be directly assigned to the CPU. In the case of a GPU, the thread has to use a library like OpenGL or DirectX to get hold of the processing unit resources. Now there are languages like CUDA, which helps program the GPU directly, which is still evolving.

Q: Could you explain more about hyper threading?

A: Hyper threading is a feature where the OS is tricked into believing that there is more than one processor for each core, therefore allowing a machine which is HT enabled to support twice the number of threads with the same CPU infrastructure.

Q: Any multithreading optimization tools from Intel?

A: Yes, we have Intel® C++ Compilers and Intel® Fortran Compilers and Intel® Performance Libraries (IPP and MKL). We have a set of threading tools Intel® Thread Checker and Intel® Thread profiler and Intel® VTune™ to find threading errors and to tune your threaded applications. Please refer <http://intel.com/software> for more

Q: Any Linux specific multithreading tools from Intel?

A: Intel® Compiler, Intel® Performance Libraries, Intel® VTune™, Intel® Thread Checker and Intel® TBB all support Linux\*. Please visit this site to access 'System requirement Document' <http://www3.intel.com/cd/software/products/asmo-na/eng/index.htm>

Q: Could you please explain more about OpenMP threads?

A: OpenMP is a Standard. Intel® Compilers support this standard. OpenMP is set of Pragmas, APIs and Flags. For more information about OpenMP visit [www.openmp.org](http://www.openmp.org)  
For more information about Intel® Compiler support to OpenMP, please visit <http://www3.intel.com/cd/software/products/asmo-na/eng/index.htm>

Q: Is it right that software synchronization is possible only if your hardware synchronization primitives are available for use?

A: No they are totally unrelated

Q: What are the pros and cons of multiprocessor systems against a multithreaded processor system?

A: In many cases, the applications can be divided into multiple tasks that also can execute in parallel. Multithreading extends the idea of multitasking to apply within applications, giving applications the capability to separate their own tasks into individual threads.

Q: Does Intel Core i7 platform have hardware memory coherency for L2 and L3 caches

A: Yes, there is a memory coherency protocol that is used for L2 and L3 caches. This ensures that the memory is coherent across the system and across all threads. MESI is one such protocol.

Q: Is Intel Core i7 platform developed to be used as a networking processor?

A: No it is not. The Intel® IXP Network processor family is meant for that.

Q: What is a pthread?

A: pthread is programming API available in the OS (typically used on Linux) for writing threaded programming.

Q: What is the bandwidth of FSB for Intel Core i7 platform? How many cores can it support?

A: Answer depends on type of mother board. Quick path can deliver about up to 6.4 GT/Sec. Refer to Intel® Core™ i7 wiki...It gives you a nice overview. Intel® Core™ i7 can support up to 4 cores today and each core supports 2 threads.

Q: There are some other coherency protocols other than MESI. Why are they required? e.g moesi

A: A protocol is a way of hand-shake. Each vendor or manufacturer would use what is applicable to their design. Hence we have MESI and MOESI protocols as required by the architecture.

Q: What is difference between semaphore and mutex?

A: Semaphore is a protected variable or abstract data type which constitutes the classic method for restricting access to shared resources such as shared memory in a multiprogramming environment. A mutex is a common name for a program object that negotiates mutual exclusion among threads

Q: What are pros and cons of multiprocessor systems against a multithreaded processor system?

A: Some main advantages of multi processor systems are given below.

- (1) As we increase the number of processors it means that more work can be done in less time. If more than one processor cooperates on a task then they will take less time to complete it.
- (2) If we divide functions among several processors, then if one processor fails then it will not affect the system or we can say it will not halt the system, but it will affect the work speed. Suppose I have five processors and one of them fails due to some reason, then each of the remaining four processors will share the work of the failed processor. So it means that system will not fail but definitely the failed processor will affect the speed.

Q: What are basic hardware primitives used for synchronization in Intel Core i7, e.g like lwarx stawx instructions

A: Hardware primitives for synchronization are used to improve performance of locking and unlocking and sharing of resources. Thus instructions like lwarx are useful to efficiently multi-threading at the hardware and low-levels.

Q: What is difference between MESI and MOESI?

A: MESI acronym expansion has been answered before. MOESI adds another state which is O- Owner. These are different protocols used by the vendors as required.

Q: What about multithreading concept in a web chat?

A: Multithreading concept is very much applicable in web chat as well. For eg: Chat window is on one thread, audio on other thread, Content sharing on other thread and so on. There are several other things that can happen in parallel apart from the ones listed above. I hope this answers your questions.

Q: What is the future of multithreading?

A: Multithreading is the basis for getting efficient performance out of many core system. From compute architecture perspective, multi-core/ many cores are here to stay. Hence threading is going to be an important architectural feature for software design.

Q: What is hyper threading?

A: Hyper threading is a way to improve the utilization of CPU, where 2 threads can be scheduled at the same time and when one thread is waiting for any I/O or data from memory, the other thread is executed. Most of the server applications can take advantage of hyper threading. Any client application which is threaded may benefit from hyper threading. Hyper threading helps minimize memory latency by running the other thread when the first thread is waiting on a resource. For single threaded applications you may not see a benefit from hyper threading

Q: Is there any relation to multithreading and paging?

A: Multi-threading is for creating programs which use multiple threads for execution of a given task or program. Thus giving benefit in performance on a multi-core and multi-processor machine.

Q: what is different between threads used by Microsoft and Mac?

A: Microsoft\* uses Win32\* threads and Mac\* uses pthreads. Do you have a specific question on this?

Q: For programming purpose which processor should I use?

A: Core micro architecture based products are the current mainstream line of Intel processors. We would recommend the usage of any of the Core based products for development. Intel® Core™ 2 Duo, Intel® Core™ 2 Quad, Intel® Core™ i7 or Intel® Core™ 2 Extreme are the ones that can be used for development

Q: If I want save this chat, can I??

A: The transcript of the chat will be archived and available to you at <http://software.intel.com/en-us/articles/web-chat-series> Please book mark this page for future reference.

Q: If I want to sync all threads at a certain point in time what barrier mechanism can I use such that it is simple and easy to use and debug too?

A: Some of the programming languages and APIs support synchronizing all threads at a certain point. For example Win32\* supports waitformultipleobjects API which allows to sync multiple threads at one point. Similarly OpenMP supports barrier for sync up of multiple threads. So please check the programming language API for details.

Q: If I need to stop a process having multiple threads, from another process (possibly a parent making a child process with multiple threads wait), what pthread API can I use?

A: There is no API that will stop the execution of all the threads. Instead you can use condition variables for synchronization among these threads. Or using the condition variables, you can broadcast the required information to all the threads.

Q: What is the difference between Centrino and Centrino2 technologies?

A: Intel® Centrino® 2 is faster & more recent than Intel® Centrino®. "Centrino" is the Intel Brand for Mobile Devices, which is a combination of Processor, Chipset and WiFi. A good comparison on the two can be found at <http://en.wikipedia.org/wiki/Centrino> Read more about the products at <http://www.intel.com/products/index.htm>.

Q: Tell us about some best practices to be followed while multithreading

A: There are many best known methods to implement threading in your application. Intel experts have written some good articles that will help you to understand multithreading. Please visit Our Knowledge Base. <http://software.intel.com/en-us/articles/multi-core/all/1/>

Q: Can you explain multi-process vs. multithreading?

A: Multitasking is running multiple "heavyweight" processes (tasks) by a single OS.

Multithreading is running multiple "lightweight" processes (threads of execution) in a single process / task / program. See *What is the difference between a lightweight and a heavyweight process in the web?* for more detail on lightweight vs. heavyweight processes.

Multiprogramming is essentially a synonym for multitasking (though multitasking connotes sharing more resources than just the CPU, and is the more popular term).

Q: I have designed my multithreading program for single core processors. Now, will the same program run fine on a multi core processor without any changes?

A: In short yes. If the application is designed to handle concurrent execution in a robust fashion, then it will run without any new bugs on the multi-core processor. In general, such applications will see performance improvements to various extent depending upon the how well parallel execution is designed. Also, it is a good idea to validate (test) the application on a multi-core processor before the product release, as sometime the code may have certain threading related bugs, which may show-up while running on multi core CPU.

Q: What's next after multi-core?

A: The multi-core revolution has just begun, we have moved from single core to now 4 cores and moving on to 6 cores across platforms. What has also been happening in parallel is the switch in manufacturing technology from 65 NM, to 45 NM to 32 NM which allows for more and more features to be integrated into the CPU, good example is audio support and now graphics and video support.

Q: Can you please brief on multi-process vs. multi threading

A: Short answer is: threads are lightweight, programs (aka processes or tasks) are heavyweight

Q: In Java, when 2 threads are in execution, what does the current thread return?

A: Depends on what the current thread was implementing

Q: WaitAll() and WaitOne() method in C# behaves very much different for Win XP\* and WIN Server 2003? How to handle it across OS?

A: waitall:-- Waits for all the elements or threads in the specified array to receive a signal, using an Int32 value to measure the time interval and specifying whether to exit the synchronization domain before the wait.

waitone:-- Blocks the current thread until the current WaitHandle receives a signal, using a 32-bit signed integer to measure the time interval and specifying whether to exit the synchronization domain before the wait.

Q: Can you please brief me about Intel's Multithreading tool?

A: Use the Intel® Thread Profiler collector to help you identify and locate bottlenecks that are limiting the parallel performance of your multi-threaded application and you can also look at Intel Thread checker for looking at functionality issues (data races etc).

Q: Can you please brief me about Intel's Multithreading tool?

A: we have threadchecker,threadprofiler 2-tools

Intel® Thread Checker to locate bugs in your multi-threaded programs. Thread Checker identifies diagnostics by finding places where the behavior of your program in a multi-threaded environment is potentially non-deterministic.

Q: Is multithreading better than multiprocessing. If so, can u please explain how?

A: Multiprocessing is the way of handling multiple processes at one point. Whereas multithreading is the concept where multiple threads get executed in parallel. Multithreading is more efficient in the sense that memory is being shared among multiple threads, whereas each process has its own memory space. Synchronization becomes a bigger problem in case of multiprocessing and context switching is very expensive in case of multiprocessing.

Q: What is the difference between UI thread & worker thread?

A: The difference between UI thread and Worker thread is as follows: I will try to demonstrate this with an example. For eg, in MS Word, the document where you type your content, is running on an UI thread. If you do a spell check, that is being performed by a worker thread that runs in parallel with the UI thread. Hope this answers your question.

Q: What are the different types of threads in C++?

A: You can use Win32\* threads on Windows, pthreads on Linux and also you can use OpenMP to thread your C++ app. You can also use Intel® Threading Building Blocks which give you better performance. Intel® Compilers support the latest OpenMP standard. Please refer [www.openmp.org](http://www.openmp.org) and [www.threadingbuildingblocks.org](http://www.threadingbuildingblocks.org) to know more on these.

Q: What actually is IPC

A: IPC means inter process communication or Instructions per Clock. Could you please clarify? Inter process communication means communication between processes on same machine or distributed machines. Instructions per clock on the other hand would mean number of instructions that are executed per clock. We would want it to be a low value on a given processor for better performance.

Q: I have an Intel Core 2 Duo processor and Windows XP Professional OS. Which OS is better for multi threading - XP or Vista?

A: Both of them support threading, it depends on the software you are planning to use

Q: Performance wise which OS is better Windows XP Pro / Vista?

A: Depends on the usage model.

Q: If there is any special tools available for VS 2005/2008 for multi threading?

A: Recently Intel came up with a new tool set called Intel® Parallel Studio. This is specifically for C/C++ developers using Microsoft Visual Studio. Please participate in Beta Program at [www.intel.com/go/parallel](http://www.intel.com/go/parallel)

Q: My Intel Core 2 Duo Processor is getting over heated as it reaches 90°C when I debug my N-Tier web application. The fan is still on. Can I know why the CPU is over heated? The same application is debugged fine with out the CPU over heating than 60°C in P4 620 processor.

A: It is strange problem. Have you checked with the supplier from whom you bought this machine, if they can debug the issue with your machine?

Q: What are different types of multithreading?

A: There are many implementation of threading. Examples include POSIX\*, Win32\* threading models.

Different OSs have different implementations. POSIX\* is a well accepted so is the Win32\* model.

Q: Does DOS supports multithreading?

A: Yes. It does. Many Libraries exist to help you thread on DOS.

Q: What are the draw backs of multi threading?

A: There may be threading overheads that affect performance. This can happen if the problem size is too small. Multithreading may lead to threading issues like deadlocks, data races, synchronization issues, if you don't load balance your threads, your application may not scale.

Q: Can we use multiple threads without giving them priority?

A: Yes, you can use multiple threads without giving any specific priority. The OS assigns default priority to the threads.

Q: Thread.sleep

A: Blocks the current thread for the specified number of milliseconds. Sleep(Int32) - Suspends the current thread for a specified time. Sleep(TimeSpan) - Blocks the current thread for a specified time.

Q: Tell me about deadlock

A: Deadlock is a common problem in multiprocessing where many processes share a specific type of mutually exclusive resource known as a software or soft lock. But if the locks do not get released successfully it would cause a resource block, forcing threads to wait.

Q: What is the main use of threads in core Java?

A: The main use of threading in Java is to facilitate multi-threading within Java. It provides an easy mechanism by which a Java developer can easily multi-thread their Java code

Q: How does threading working in web?

A: Performance and responsiveness are important factors in the success of your web application. Users quickly tire of using even the most functional application if it is unresponsive or regularly appears to freeze when the user initiates an action. Even though it may be a back-end process or external service causing these problems, it is the user interface where the problems become evident. Multithreading and asynchronous programming techniques enable you to overcome these difficulties. Application performance and scalability can be greatly enhanced using multithreading and asynchronous operations. Wherever possible, try to use these techniques to increase the responsiveness of your presentation layer components.

Q: What is the difference between a thread and a process?

A: A process and a thread are a stream of instructions that are executed in the processor. However, each thread has its own stack but shares the same memory space along with other threads. However, a process has a unique memory space that is exclusive to each process.

Q: How to resolve deadlock?

A: To resolve deadlock, you have to use Synchronization objects. It may be Locks, Mutex etc. Intel® Threading Building Block provides you many Synchronization objects, Please visit, <http://www3.intel.com/cd/software/products/asmo-na/eng/index.htm> to download Intel® TBB

Q: Do you have any specific processors for automotive industry?

A: Intel Embedded Products powers embedded designs in automotive industry. To see how the smallest Intel Embedded chips enable designs, please look at the flash demo on

<http://www.intel.com/go/soc/> To see how embedded powers a concept Intel Embedded motorcycle look at <http://www.intel.com/pressroom/archive/releases/20070403corp.htm> To know how to develop in-car entertainment, see <http://www.intel.com/pressroom/archive/reference/IVIOpenPlatforms.pdf>

Q: What approach should be taken while designing a multi threaded program?

A: There are some papers posted on the Intel® Software Network on design aspects of multi threaded programs which you could have a look at [www.intel.com/software](http://www.intel.com/software) . In general, the guidelines are to look for threaded performance issues, coding and correctness issues and consider the end platform on which the code is executed.

Q: You said Intel Core i7 has HT capability. So what is the cache size required in that processor?

A: Intel® Core™ i7 comes with different cache sizes. 64 KB L1 instruction and 64 KB L1 data cache per core, 256 KB L2 cache (combined instruction and data) per core, 8 MB L3 (combined instruction and data) "inclusive", shared by all cores. Were you thinking Memory size when you said Cache?

Q: What are cons of HT?

A: What happens in HT is that the same execution unit/cache is being shared between multiple threads. For application which requires huge amount of cache it may be at a disadvantage because multiple threads are sharing the cache.

Q: Is software threading bounded by processor architecture?

A: The number of cores and threads supported has been increasing with the introduction of every new architecture. Beyond that it does not matter. If you implement scalable threading, then your software should be able to scale as the number of cores increases from one architecture to another.

Q: Is there any difference in multithreading capability of Intel Core2Duo processor and dual core processor?

A: Both the processors are dual core, they differ in architecture and implementation of multi-core architecture. In addition to moving from 65 nm to 45 nm and to now 32 nm, a dual core processor will support 2T. The performance will improve from the older dual core processor to the newer Intel® Core™ 2 Duo.

Q: Is hardware multithreading strong or software multithreading?

A: Hardware multithreading makes very efficient utilization of resources. For e.g: Intel Hyper Threading makes efficient use of the processor resources. Software multithreading provides the developer the ability to utilize multi core /multiprocessing capabilities of the platform efficiently.

Q: Can you please tell me if over-clocking will harm my processor?

A: Most of the processors come with an upper limit for clock speed which is designed to protect the processor from damage. The new Intel® Core™ i7 processor launched in 2008, comes with a built in over-clocking feature which allows the user to step up the clock speed. It also has an automatic over-clocking feature. In spite of this there are limits to what can be achieved in a normal room temperature environment. The max I have recorded is around 4-5 GHZ on an Intel® Core™ i7 but that was with special cooling.

Q: May multithreading cause any harm to processor?

A: No physical harm to processor or damage

Q: Is there a tool which can let me know the possible deadlocks in code during compile time?

A: No, it is not possible to find the possible deadlocks in your code during compile time. You can use

Intel® Thread Checker or our new Intel® Parallel Studio which is in Beta to find it during run time and fix it. Please find more info on Intel® tools at <http://www.intel.com/software>

Q: I want to know multithreading in Intel Core 2 Duo?

A: Multithreading is a popular programming and execution model that allows multiple threads to exist within the context of a single process. These threads share the process' resources but are able to execute independently. The threaded programming model provides developers with a useful abstraction of concurrent execution. However, perhaps the most interesting application of the technology is when it is applied to a single process to enable parallel execution on a multi-core system such as an Intel® Core™ 2 Duo processor.

Q: What are Mutex threads?

A: Mutex is a concept in threading, where in it is used as a synchronization object to ensure that there are no threading errors or data races. Mutex is a lock object for ensuring exclusive access to objects and code for each thread.