1. What’s the difference between a thread and a process?

Process and thread share a relationship where a process provides an environment for the execution of the thread. A process can contain multiple threads.

Comparison Chart

| **BASIS FOR COMPARISON** | **PROCESS** | **THREAD** |
| --- | --- | --- |
| Basic | An executing program is called a process. | A thread is a small part of a process. |
| Address Space | Every process has its separate address space. | All the threads of a process share the same address space cooperatively as that of a process. |
| Multitasking | Process-based multitasking allows a computer to run two or more than two programs concurrently. | Thread-based multitasking allows a single program to run two or more threads concurrently. |
| Communication | Communication between two processes is expensive and limited. | Communication between two threads is less expensive as compared to process. |
| Switching | Context switching from one process to another process is expensive. | Context switching from one thread to another thread is less expensive as compared to process. |
| Components | A process has its own address space, global variables, signal handlers, open files, child processes, accounting information. | A thread has its own register, state, stack, program counter. |
| Substitute | Process are also called heavyweight task. | Thread are also called lightweight task. |
| Control | Process-based multitasking is not under the control of Java. | Thread-based multitasking is under the control of Java. |
| Example | You are working on text editor it refers to the execution of a process. | You are printing a file from text editor while working on it that resembles the execution of a thread in the process. |

1. Context Switch: How would you measure the time spent in a context switch?

A *Context switch* is the time spent between two processes (i.e., bringing a waiting process into execution and sending an executing process into waiting state). This happens in multitasking.

6. Synchronized Methods: You are given a class with synchronized method A and a normal method B. If you have two threads in one instance of a program, can they both execute A at the same time?

Can they execute A and B at the same time?

No they cannot execute A at same time, yes but they can execute A and B at same time.

8. How a thread can interrupt another thread in Java? Show with an example.

**Interrupt() method:**If any thread is in sleeping or waiting state then using interrupt() method, we can interrupt the execution of that thread by showing InterruptedException. A thread which is in the sleeping or waiting state can be interrupted with the help of interrupt() method of Thread class.

// Java Program to illustrate the

// concept of interrupt() method

// while a thread does not stops working

class MyClass extends Thread {

    public void run()

    {

        try {

            for (int i = 0; i < 5; i++) {

                System.out.println("Child Thread executing");

                // Here current threads goes to sleeping state

                // Another thread gets the chance to execute

                Thread.sleep(1000);

            }

        }

        catch (InterruptedException e) {

            System.out.println("InterruptedException occur");

        }

    }

}

class Test {

    public static void main(String[] args)

              throws InterruptedException

    {

        MyClass thread = new MyClass();

        thread.start();

        // main thread calls interrupt() method on

        // child thread

        thread.interrupt();

        System.out.println("Main thread execution completes");

    }

}