BENEFITS OF MULTITHREADING AND PROCESS VS THREAD

Why Multithreading:

In certain situations, a single application may be required to perform several similar task such as a web server accepts client requests for web pages, images, sound, graphics etc. A busy web server may have several clients concurrently accessing it. So if the web server runs on traditional single threaded process, it would be able to service only one client at a time. The amount of time that the client might have to wait for its request to be serviced is enormous.

One solution of this problem can be thought by creation of new process. When the server receives a new request, it creates a separate process to service that request. But this method is heavy weight. In fact this process creation method was common before threads become popular. Process creation is time consuming and resource intensive. If the new process perform the same task as the existing process, why incur all that overhead? It is generally more efficient for one process that contains multiple threads to serve the same purpose. This approach would multithreaded the web server process. The server would cerate a separate thread that would listen for clients requests. When a request is made, rather than creating another process, it will create a separate thread to service the request.

Benefits of Multi-threading:

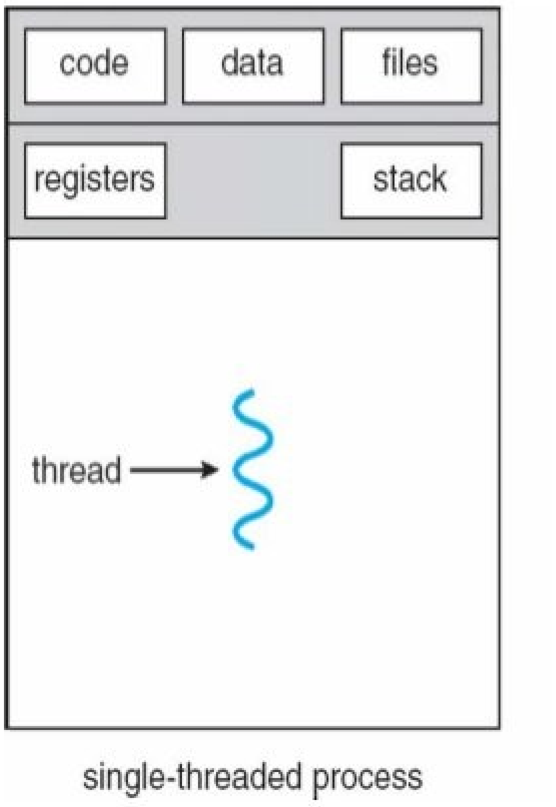
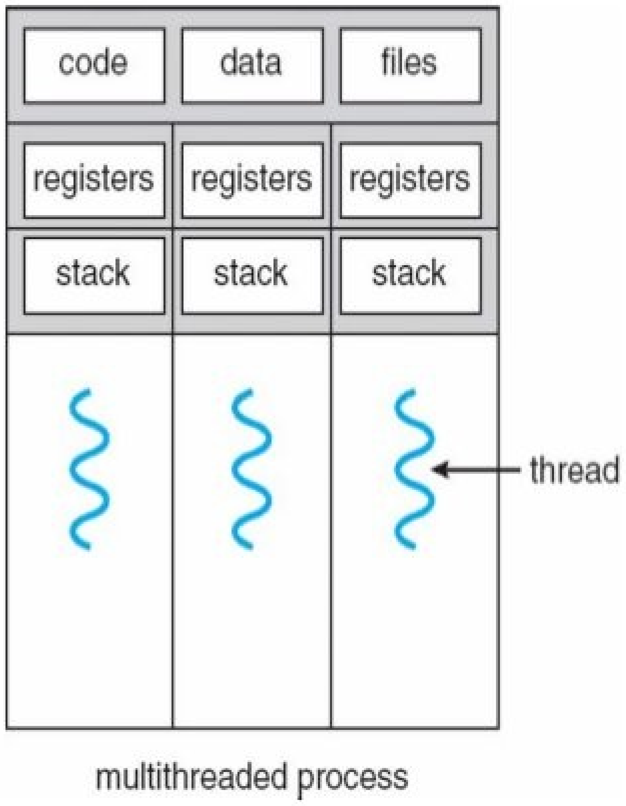
Responsiveness: Mutlithreaded interactive application continues to run even if part of it is blocked or performing a lengthy operation, thereby increasing the responsiveness to the user.

Resource Sharing: By default, threads share the memory and the resources of the process to which they belong. It allows an application to have several different threads of activity withing the same address space.

Economy: Allocating memory and resources for process creation is costly. Since thread shares the resources of the process to which they belong, it is more economical to create and context switch threads. It is more time consuming to create and manage process than threads.

Utilization of multiprocessor architecture: The benefits of multi threading can be greatly increased in multiprocessor architecture, where threads may be running in parallel on different processors. Mutlithreading on a multi-CPU increases concurrency.

Process VS Thread:

Process	***Thread***
 code data files registers stack thread single-threaded process	 code data files registers registers registers stack stack stack thread multithreaded process
Program in execution.	Basic unit of CPU utilization.
Heavy weight	Light weight
Unit of Allocation – Resources, privileges etc	Unit of Execution – PC, SP, registers PC—Program counter, SP—Stack pointer
Inter-process communication is expensive: need to context switch Secure: one process cannot corrupt another process	Inter-thread communication cheap: can use process memory and may not need to context switch Not secure: a thread can write the memory used by another thread
Process are Typically independent	Thread exist as subsets of a process
Process carry considerable state information.	Multiple thread within a process share state as well as memory and other resources.
Processes have separate address space	Thread share their address space
processes interact only through system-provided inter-process communication mechanisms.	Context switching between threads in the same process is typically faster than context switching between processes.

Source : http://dayaramb.files.wordpress.com/2012/02/operating-system-pu.pdf