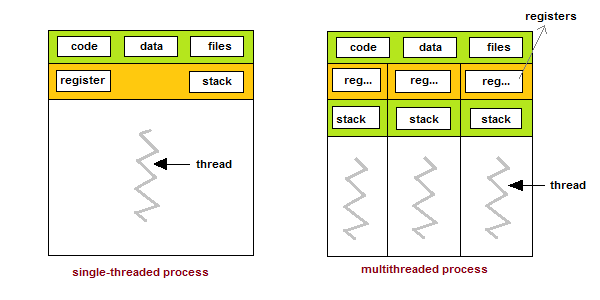
INTRODUCTION OF THREADING

Thread is an execution unit which consists of its own program counter, a stack, and a set of registers. Threads are also known as Lightweight processes. Threads are popular way to improve application through parallelism. The CPU switches rapidly back and forth among the threads giving illusion that the threads are running in parallel.

As each thread has its own independent resource for process execution, multpile processes can be executed parallely by increasing number of threads.

DEFINITION OF MULTI-THREADING

Multithreading is a type of execution model that allows multiple threads to exist within the context of a process such that they execute independently but share their process resources. A thread maintains a list of information relevant to its execution including the priority schedule, exception handlers, a set of CPU registers, and stack state in the address space of its hosting process. Multithreading is also known as threading.

Benefits of Multi-Threading in Operating System

The benefits of multi threaded programming can be broken down into four major type.

1. **Responsiveness –**  
   Multithreading in an interactive application may allow a program to continue running even if a part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user.
2. **ResourceSharing –**  
   Processes may share resources only through techniques such as-

* Message Passing
* Shared Memory

Such techniques must be explicitly organized by programmer. However, threads share the memory and the resources of the process to which they belong by default.  
The benefit of sharing code and data is that it allows an application to have several threads of activity within same address space.

1. **Economy –**  
   Allocating memory and resources for process creation is a costly job in terms of time and space.  
   Since, threads share memory with the process it belongs, it is more economical to create and context switch threads. Generally much more time is consumed in creating and managing processes than in threads.
2. **Scalability –**  
   The benefits of multi-programming greatly increase in case of multiprocessor architecture, where threads may be running parallel on multiple processors. If there is only one thread then it is not possible to divide the processes into smaller tasks that different processors can perform.

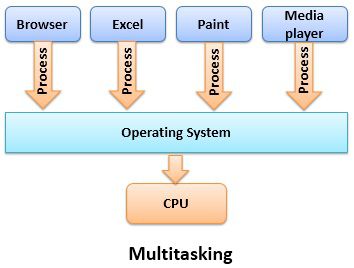
DISADVANTAGE OF MULTI-THREADING

1. synchronization of shared resources(object,data)
2. difficult to debug, result is sometimes unpredictable
3. potential deadlocks
4. “starvation”:some threads may not be served with a bad design
5. Constructing and synchronizing thread is CPU/memory intensive

### Definition of Multi-Tasking

Multitasking is when a single CPU performs **several tasks (program, process, task, threads)** at the same time. To perform multitasking, the CPU switches among theses tasks very **frequently** so that user can interact with each program simultaneously.

In a multitasking operating system, several users can **share the system** simultaneously. As we saw the CPU rapidly switches among the tasks, so a little time is needed to switch from one user to the next user. This puts an impression on a user that entire computer system is dedicated to him.



When several users are sharing a multitasking operating system, CPU **scheduling** and **multiprogramming** makes it possible for each user to have at least a small portion of Multitasking OS and let each user have at least one program in the memory for execution.

# Difference Between Multitasking and Multithreading



The basic difference between Multitasking and multithreading is that **Multitasking** allows  CPU to perform multiple tasks (program, process, task, threads) simultaneously whereas, **Multithreading** allows multiple threads of the same process to execute simultaneously. Let us discuss the differences between Multitasking and Multithreading with the help of comparison chart shown below.

| **Parameter** | **Multiprocessing** | **Multithreading** |
| --- | --- | --- |
| Basic | Multiprocessing helps you to increase computing power. | Multithreading helps you to create computing threads of a single process to increase computing power. |
| Execution | It allows you to execute multiple processes concurrently. | Multiple threads of a single process are executed concurrently. |
| CPU switching | In Multiprocessing, CPU has to switch between multiple programs so that it looks like that multiple programs are running simultaneously. | In multithreading, CPU has to switch between multiple threads to make it appear that all threads are running simultaneously. |
| Creation | The creation of a process is slow and resource-specific. | The creation of a thread is economical in time and resource. |
| Classification | Multiprocessing can be symmetric or asymmetric. | Multithreading is not classified. |
| Memory | Multiprocessing allocates separate memory and resources for each process or program. | Multithreading threads belonging to the same process share the same memory and resources as that of the process. |
| Pickling objects | Multithreading avoids pickling. | Multiprocessing relies on pickling objects in memory to send to other processes. |
| Program | Multiprocessing system allows executing multiple programs and tasks. | Multithreading system executes multiple threads of the same or different processes. |
| Time taken | Less time is taken for job processing. | A moderate amount of time is taken for job processing. |

### Comparison Chart

| **BASIS FOR COMPARISON** | **MULTITASKING** | **MULTITHREADING** |
| --- | --- | --- |
| Basic | Multitasking let CPU to execute multiple tasks at the same time. | Multithreading let CPU to execute multiple threads of a process simultaneously. |
| Switching | In multitasking CPU switches between programs frequently. | In multithreading CPU switches between the threads frequently. |
| Memory and Resource | In multitasking system has to allocate separate memory and resources to each program that CPU is executing. | In multithreading system has to allocate memory to a process, multiple threads of that process shares the same memory and resources allocated to the process. |