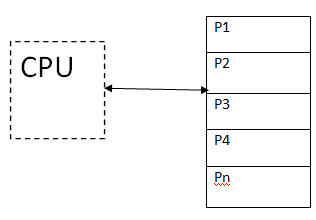
JavaTpoint:

Multiprogramming vs Multiprocessing vs Multitasking vs Multithreading

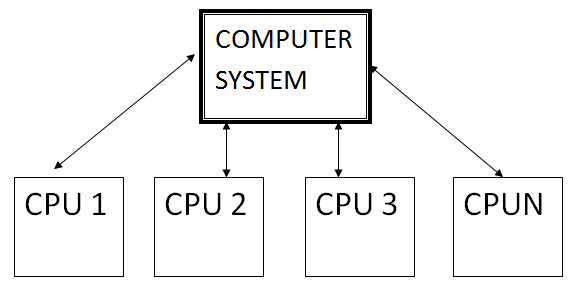
Multiprogramming

* "The concurrent residency of more than one program in the main memory is referred as multiprogramming."
* Since multiple programs are resident in the memory, as soon as the currently executing program finishes its execution, the next program is dispatched for its consumption.
* Also if the currently executing program asks for input output resources then meanwhile another program is dispatched to the CPU for execution.
* The main objective of multiprogramming is:
  + Maximum CPU utilization.
  + Efficient management of the main memory.
* Multiprogramming can be virtually shown as:



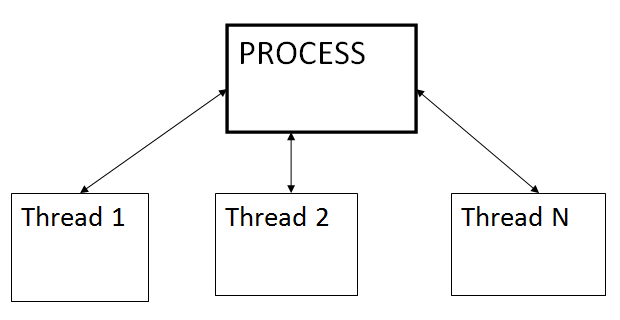
Multiprocessing

* When one system is connected to more than one processor which collectively work for the completion of the task, it is called as multiprocessing systems.
* Multiprocessing systems can be divided in two types:
  + **Symmetric Multiprocessing:** The operating system here resides on one processor and the other processors run user's programs.
  + **Asymmetric Multiprocessing:** The OS runs on any available processor or all the processor simultaneously run the user program.
* Multiprocessing systems can be virtually represented as:



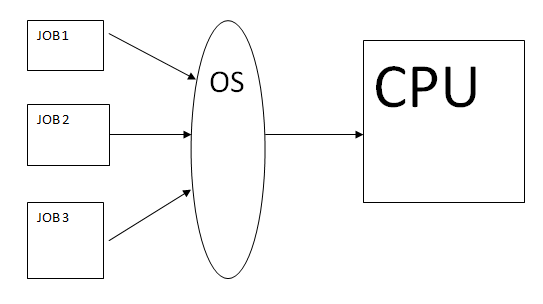
Multithreading

* "Multithreading is a conceptual programming paradigm where a process is divided into a number of sub-processes called as threads. Each thread is independent and has its own path of execution with enabled inter thread communication."
* "Thread is the path followed while executing a program. Each thread has its own program counter, stack and register."
* A thread is a light weight process.
* It can be virtually represented as:



Multitasking

* Earlier when computers were invented, a user was allowed to submit only job or task at a time. But later with availability of high-speed processor, one can submit more than one task.
* So the capability of OS to accept more the one task per user is termed as multitasking.
* Multiple jobs are executed by the CPU simultaneously by switching between them.
* The various job can be accepted from same user or different users. There are 2 types of multitasking systems:
  + **Single User Multitasking**
  + **Multi User multitasking**
* It can be virtually represented as:



**Now, let's take a look at difference between these types of system:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Characteristic** | **Multiprogramming** | **Multiprocessing** | **Multithreading** | **Multitasking** |
| 1 | What it is: | The concurrent residency of more than one program in the main memory is called as multiprogramming. | The availability of more than one processor per system, which can execute several set of instructions in parallel is called as multiprocessing. | A process is divided into several different sub-processes called as threads, which has its own path of execution. This concept is called as multithreading. | The execution of more than one task simultaneously is called as multitasking. |
| 2 | Number of CPU: | One | More than one | Can be one or more than one | One |
| 3 | Job processing time: | More time is taken to process the jobs. | Less time is taken for job processing. | Moderate amount of time is taken for job processing. | Moderate amount of time. |
| 4 | Number of process being executed: | One process is executed at a time. | More than one process can be executed at a time | Various components of the same process are being executed at a time. | One by one job is being executed at a time. |
| 5 | Economical: | It is economical. | Is less economical. | Is economical. | It is economical. |
| 6 | Number of users: | One at a time. | Can be one or more than one. | Usually one. | More than one. |
| 7 | Throughput: | Throughput is less. | Throughput is maximum. | Moderate. | Throughput is moderate. |
| 8 | Efficiency: | Less | Maximum | Moderate | Moderate |
| 9 | Categories: | No further divisions | Symmetric & Asymmetric. | No further divisions. | Single User & Multiuser. |

Greekforgreeks:

1. Multiprogramming – Multiprogramming is known as keeping multiple programs in the main memory at the same time ready for execution.
2. Multiprocessing – A computer using more than one CPU at a time.
3. Multitasking – Multitasking is nothing but multiprogramming with a Round-robin scheduling algorithm.
4. Multithreading is an extension of multitasking.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Multiprogramming | Multitasking | Multithreading | Multiprocessing |
| Definition | Running multiple programs on a single CPU | Running multiple tasks (applications) on a single CPU | Running multiple threads within a single task (application) | Running multiple processes on multiple CPUs (or cores) |
| Resource Sharing | Resources (CPU, memory) are shared among programs | Resources (CPU, memory) are shared among tasks | Resources (CPU, memory) are shared among threads | Each process has its own set of resources (CPU, memory) |
| Scheduling | Uses round-robin or priority-based scheduling to allocate CPU time to programs | Uses priority-based or time-slicing scheduling to allocate CPU time to tasks | Uses priority-based or time-slicing scheduling to allocate CPU time to threads | Each process can have its own scheduling algorithm |
| Memory Management | Each program has its own memory space | Each task has its own memory space | Threads share memory space within a task | Each process has its own memory space |
| Context Switching | Requires a context switch to switch between programs | Requires a context switch to switch between tasks | Requires a context switch to switch between threads | Requires a context switch to switch between processes |
| Inter-Process Communication (IPC) | Uses message passing or shared memory for IPC | Uses message passing or shared memory for IPC | Uses thread synchronization mechanisms (e.g., locks, semaphores) for IPC | Uses inter-process communication mechanisms (e.g., pipes, sockets) for IPC |

**1**