* **Name: Abhikumar V. Patel**

|  |
| --- |
| Sp.: Parallel and distributed processing - lecture Lecturer: Bartłomiej Kotyra, Ma |
|

**Essay Topic:**

**Concurrent / Parallel Programming**

**A system is said to be concurrent if it can support two or more actions in progress at the same time. A system is said to be parallel if it can support two or more actions executing simultaneously. This definition says that, in concurrent systems, multiple actions can be in progress (may not be executed) at the same time. Meanwhile, multiple actions are simultaneously executed in parallel systems. In fact, concurrency and parallelism are conceptually overlapped to some degree, but "in progress" clearly makes them different.**

**Concurrency is when two tasks can start, run, and complete in overlapping time periods. It doesn't necessarily mean they'll ever both be running at the same instant. Exp. multitasking on a single-core machine. Parallelism is when tasks literally run at the same time, exp. on a multicore processor. The term Parallelism refers to techniques to make programs faster by performing several computations in parallel. This requires hardware with multiple processing units. In many cases the sub-computations are of the same structure, but this is not necessary. Graphic computations on a GPU are parallelism.**

**Threads are a way for a program to divide itself into two or more simultaneously running tasks. Threads and processes differ from one operating system to another but, in general, a thread is contained inside a process and different threads in the same process share same resources while different processes in the same multitasking operating system do not. the threads are what allow your CPU to perform multiple things at once. So if you want to run multiple processes that are very intensive, you will need a CPU with a lot of threads.**

**Threads refer to the highest level of code executed by a processor, so with many threads, your CPU can handle several tasks at the same time. The number of threads you have depends on the number of cores in your CPU. Each CPU core can have two threads. So a processor with two cores will have four threads. Threads are important to the function of your computer because they determine how many tasks your computer can perform at any given time.**

**A multicore processor is a single processor chip that has more than one processor on a single chip contained in a single package. A processor sometimes referred to as a "core," is a circuit that performs instructions or calculations. Since a multicore processor has more than one processing unit, it can perform calculations and run programs at faster speeds than a single processor chip. Multicore processors are commonly used in many of the current computers, smartphones and tablet devices and make our devices run faster than they would with a single core processor chip.**

**If you want to increase the performance of your program one possible** solution is to add concurrent programming techniques. Basically, in concurrent **execution, multiple threads of the same program executes at the same time. It is similar to adding more workers to complete a job. “Parallelism” is another word which used often with concurrency. Parallelism is a subset of concurrency. Concurrency refers to trying to do multiple things at once while parallelism refers to doing many things at once.**

**You've probably wondered why that is. Perhaps you've accepted the common fallacy that "Multithreading is hard." It's not. If a multithreaded program is unreliable it's most likely due to the same reasons that single-threaded programs fail: The programmer didn't follow basic, well known development practices. Multithreaded programs seem harder or more complex to write because two or more concurrent threads working incorrectly make a much bigger mess a whole lot faster than a single thread can.**