Task M2.S1P

Activity 1.

1. Definition:
   - Concurrent programming: is in which, during a period of time, multiple process are being executed. A concurrent program is a program defines multiple actions/statements that may performed simultaneously on one or more core (CPU). Example: Deposit and Withdraw money.
   - Parallel programming: is the process of breaking down large problem into multiple smaller, independent problems. Those smaller multiple problems are usually similar parts that can be executed simultaneously, and the outcome is the combined consists of those smaller problems. A parallel program is a concurrent program that is designed for execution on parallel hardware (Multi-core). Example: screening a video
   - Distributed programming: one in which components located at networked computers communicate and coordinate their actions only by passing messages. A distributed program is a parallel program designed for execution on a network of autonomous processors that do not share memory. Example: multiple teams on different locations work on a same project.

   Differences:

   - Concurrent: executes multiple task during a period of times but not simultaneously. No communication or coordination between processes
   - Parallel: executes multiple task simultaneously. No communication or coordination between processes
   - Distributed: executes multiple task simultaneously and communicate/coordination existed between those processes
2. Amdahl's Law is a formula that calculates the theoretical increase in latency of task execution at a certain workload that can be predicted from a system with improved resources. It can shows the fundamental limit to how fast you can make your concurrent code
3. Speedup execution of all floating-point operations by 2 times. Assume 50% of the total execution time of the program is spent in executing floating-point operations since it will increase the calculation of FPSR by 196%

Activity 2.

1. Explains and examples:
   a. Single-instruction, single data (SISD): is a uniprocessor machine which is capable of executing a single instruction, operating on a single data stream. Examples: pipelined processors and superscalar processors.
   b. Multiple-instruction, single data (MISD): is a multiprocessor machine capable of executing different instructions on different PEs but all of them operating on the same dataset. Examples: Systolic arrays
   c. Single-instruction, multiple data (SIMD): is a multiprocessor machine capable of executing the same instruction on all the CPUs but operating on different data streams. Example: Wireless MMX unit

d. Multiple-instruction, multiple data (MIMD): is a multiprocessor machine which is capable of executing multiple instructions on multiple data sets. Example: Intel Xeon Phi

2. Array Processor is a subcategory of SIMD since they consist of an array of functional units with a shared controller. Example: ILLIAC IV computer
3. SPMD is a subcategory of MIMD because tasks are split up and run simultaneously on multiple processors with different input in order to obtain results faster. Examples: the parallel DO loop

   MPMD is a subcategory of MIMD because multiple autonomous processors simultaneously operating at least 2 independent programs. Example: Sony PS3