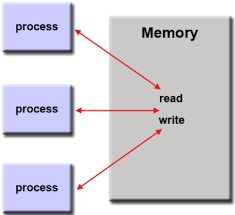
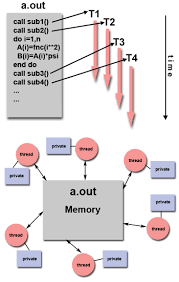
Bryanna Hardy

Task 3: Parallel Programming Skills

1. Define the following: Task, Pipelining, Shared Memory, Communications, and Synchronization.
   1. Task is a logically discrete section of computational work. It is typically a program set of instructions that is being executed by a processor.
   2. Pipelining breaks a task into steps performed by different processor units; It is kind of like an assembly line.
   3. Shared Memory describes a computer architecture where all processors have direct access to a common memory.
   4. Communications is a way for parallel tasks to exchange data.
   5. Synchronization is the coordination of parallel tasks in real time. It usually involves waiting by at least one task, and causes a parallel application’s clock execution time to increase.
2. Classify parallel computers based on Flynn’s taxonomy. Briefly describe every one of them.
   1. Flynn’s taxonomy distinguishes multi-processor computer architectures according to how they can be classified along the two independent dimensions. There are four classifications:
      1. Single Instruction/Single Data (SISM): Only one instruction stream is being acted on by the CPU during any one clock cycle. Only one data stream is being used as an input during any one clock cycle. This is also the oldest type of computer.
      2. Single Instruction/Multiple Data (SIMD): All processing units executes the same instructions at any clock cycle. Each processing unit operates on a different data element. It is best suited for specialized problems characterized by a high degree of regularity.
      3. Multiple Instruction/Single Data(MISD): Each processing unit operates on the data independently with different instruction streams. The single data stream is fed into multiple processing units.
      4. Multiple Instruction/Multiple Data(MIMD): Each processor might be executed on a different instruction stream. Every processor may be working with a different data stream. The most common type of parallel computer.
3. What are the Parallel Programming Models?
   1. Shared Memory(w/o threads), threads, distributed memory/ message passing, data parallel, hybrid, single program multiple data (SPMD), and multiple program multiple data (MPMD). They exist as an abstraction above hardware and memory architectures. They can also be implemented on any underlying hardware.
4. List and briefly describe the types of Parallel Computer Memory Architecture. What type is used by OpenMP and why?
   1. Uniform Memory Access (UMA): Commonly represented today by Symmetric Multiprocessor (SMP) machines. Have equal access and access times to memory.
   2. Non-Uniform Memory Access (NUMA): Often made by physically linking two or more SMPs. Memory access across links is slower. Also, not all processors have equal access time to all of the memories.
   3. Uniform Memory Access (UMA) is used by OpenMP because it has equal access and access times to the memory and it is also a multi-processor.
5. Compare Shared Memory Model with Threads Model? (in your own words and show pictures).
   1. In Shared Memory Model (without threads), the processes/tasks share a common address space, which reads/writes asynchronously. Semaphores are used to control the access to the shared memory.



* 1. In Threads Model, it is a type of shared memory programming. A single heavy weight process can have multiple light weight, concurrent execution paths.



1. What is Parallel Programming?
   1. Parallel Programming is the simultaneous use of multiple compute resources to solve a problem.
2. What is system on chip (SoC)? Does Raspberry PI use a system on SoC?
   1. SoC integrates the majority of the components into a single silicon chip: GPU, memory, USB controller, power management circuits, and a wireless radio. Raspberry PI does use a system on SoC because it does not have a separate CPU, RAM,or GPU, instead they are squeezed into one component, which is SoC.
3. Explain what the advantages are of having a System on a Chip rather than separate CPU, GPU, and RAM components.
   1. One of the advantages is the SoC size, it is a little bit bigger than the CPU. Also, it can consume less power, which makes it cheaper to build a computer using an SoC.