Nathan Heckman

Parallel Programming Foundations

1. **Identify the components on the Raspberry Pi B+.**

Components on the Raspberry Pi B+ include the CPU/RAM, Ethernet Controller, Ethernet, USB ports, power, HDMI, camera, and display.  
**2. How many cores does the Raspberry Pi’s B+ CPU have?**

Raspberry Pi B+ CPU has 4 cores.

1. **List three main differences between X86 (CISC) and ARM Raspberry PI (RISC).**
2. X86 uses complex instruction set computing (CISC), more features and ways to access memory compared to ARM
3. ARM uses reduced instruction set computing (RISC), more general registers for memory
4. X86 uses little-endian while ARM has the option to run on both little and big-endian modes
5. **What is the difference between sequential and parallel computation and identify the practical significance of each?**

Sequential – series of instructions, single processor, only one instruction at a time, longer

Parallel – instructions executed concurrently, multiple instructions at once, multi-core, faster

Parallel can be used in databases and virtual reality, for example. All modern computer systems support this and use it. Sequential is ok for tasks that are always the same and are fairly small.

1. **Identify the basic form of data and task parallelism in computational problems.**

Data parallelism – simultaneous execution of different functions on multiple cores across dataset elements

Task parallelism – simultaneous execution of same function on multiple cores across dataset elements

1. **Explain the differences between processes and threads.**

Processes – running program abstraction, programs do not share memory among each other

Threads – decomposed process into smaller, independent parts. Common memories shared

1. **What is OpenMP and what is OpenMP pragmas?**

OpenMP is a library language that supports multiprocessing programming on many platforms. OpenMP pragmas are compiler directives that generate threads.

1. **What applications benefit from multi-core (list four)?**

Compilers, database servers, web servers, and most applications with thread-level parallelism benefit from multi-core processing.

1. **Why Multicore? (why not single core, list four)**

Multi-core processors are easier to make compared to increasing single-core frequencies. Also, the future of application development is multithreaded, so multicores will be important to keep up with this trend in technology advancement. Parallelism is also preferred since it is quicker and can preform multiple operations at once. Multicore usage helps architects avoid technological obsolescence and improve maintainability.