1. Raspberry Pi B+ has 1gb of RAM, a quad core ARM based processor at 1.4ghz, a micro-usb power supply, 3.5mm audio out jack, HDMI out jack, Gigabit Ethernet, and 4 USB 2.0 ports, there is also a GPIO header and camera header on the motherboard.
2. The B+ has a quadcore cpu at 1.4ghz clock speed.
3. X86 (CISC) is considered to have a register memory structure whereas ARM(RISC) is a register-register implementation. CISC is considered more flexible because it allows for a more complex instruction set, however it has less registers than ARM. ARM uses Big endien notation but allows for switching between endiens. Finally ARM is considered faster in some respects because of less instructions needed to perform a task. However creative programming needs to be implemented in order to get similar overall results to that of x86.
4. Sequential computation is good for code that cannot be easily broken up into parallel chunks. Parallel is good for chunks of code that can be parallelized such as loops.
5. Data parallelism is defined as parallel calculations that can performed to compute data in parallel. Task parallelism involves not just parallel data computations but the overall set of tasks that occur can be parallelized i.e. a whole set of equations. The main distinction here is that data parallelism happens to a strict set of data whereas Task is more broad.
6. Processes are distinct instances of a program that perform a myriad of tasks, there can be sub processes in a process that allow for discrete processing however they should not be confused with threads which allow for multiple instances of code for a process to be split and be processed in parallel on multicore cpus before being joined after they output the desired result.
7. OpenMP is the framework designed for the C language that allows for implicit multi-threading and parallelism in processes. It is implicit because it allows for a pre-built framework to code and utilize multithreading through the OS itself. This function is achieved via a compiler directive called “pragmas” that signals the OS to treat the code segment as parallel. It is achieved via a single task multiple data implementation.
8. Applications that benefit heavily from multicore systems are Web browsers with multi-media plugins, web servers (where every new connection through a socket could be handled by a core on a multithreaded CPU, database servers which are also constantly working on multiple tasks at one time, and finally modern PC and console and mobile games that run on X86 and ARM architectures. Games such as Battlefield 3, and PUBG are designed from the ground up to utilize multicore cpus with 4 and 6 cores respectively to process the games real time game engines.