Task 3 Part A

Identifying the components on the Raspberry PI B+

* 4 USB ports
* HDMI port
* Ethernet port
* Micro USB connection
* Stereo output AUX port
* 1.4GHz 64-bit quad-core processor,
* Dual-band wireless LAN, Bluetooth 4.2 BLE
* Power over ethernet header
* Extended 40-pin GPIO header

How many cores does the Raspberry PI B+’s CPU have?

* 4 cores (quad core)

List three main differences between x86 (CISC) and ARM Raspberry PI (RISC). Justify your answer and use your own words.

1. RISC has a simpler ISA while x86 has a much more complicated ISA
   1. RISC is designed to take in simple instructions, which has the effect of making it easy to use peripherals. However, because the instruction design is simple it must deal with lengthier instructions, more memory, and longer instruction time.
   2. X86 has much more complicated instructions that have more flexibility. It requires more chip space.
2. Power Consumption
   1. RISC takes very little power to run, and is very power efficient. It is so power efficient it can even function without a heatsink. It has fewer transistors and relatively low speeds.
   2. X86 consumes large amounts of power compared to ARM
3. Software
   1. RISC is designed for small mobile devices such as Android and the Raspberry PI
   2. X86 is designed for bigger loads such as on a standard desktop. Bigger devices bigger load

What is the difference between sequential and parallel computation and identify the practical significance of each?

* The difference between sequential and parallel computation is that is that sequential computation processes things one by one or in a linear sequence and is all executed on a single processor. Parallel computation utilizes more than one processor to accomplish things in parallel or multiple processes at the same time via the multiple processors that are available. It would take the same problem breaking it into parts that can be solved in sequence with one another.

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

* Task Parallelism
  + Simultaneous execution of various different tasks in parallel on the different cores.
* Data Parallelism
  + Simultaneous execution of the same task in parallel with each other on the separate cores.

Explain the differences between processes and threads.

* Processes
  + Typically independent and more information
  + Each one has a different address
  + Only interact through system IPC
* Threads
  + Subsets of processes
  + Share the same address base cause in the same process
  + For smaller tasks, while threads are heavyweight

What is OpenMP and what is OpenMP pragmas?

* OpenMP is a library for parallel programming that handles thread creation and management.
* OpenMP *pragmas* compiler directives that enable the compiler to generate threaded code.

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

1. Database server
2. Scientific applications
3. compilers
4. Multi-media applications

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

* It’s difficult to make single-clock frequencies even higher.
* Many applications are now multithreaded
* Most new technology and programs are tending towards Parallelism
* Multicore processors are much more powerful and capable of more complicated tasks.