Tony Ngo – Project A2 Task 3A

1. Identifying the Components on the Raspberry PI B+
   1. CPU/RAM, Display, Power, Ethernet, 2 USB Hubs, Ethernet Controller, HDMI
2. How many cores does the Raspberry Pi’s B+ CPU have?
   1. Four
3. List three main differences between x86 (CISC) and ARM (RISC)
   1. ARM has a simplified instruction set (Reduced) versus x86 instruction set which has a more intricate set (Complex)
   2. ARM uses only instructions which you can only use registers, while x86 can use memory-based instructions.
   3. X86 processors use the Little-Indian format, which means that the processor loads memory from right to left.
4. What is the difference between sequential and parallel computation and identify the practical significance of each?
   1. In sequential computation, instructions are executed on a single processor and only ONE instruction can be executed at a time. While in parallel computation, a problem is broken into parts that can be solved concurrently on different processors. Parallel programming would be practical in scenarios where more computing power is needed (i.e. complex mathematical equations that require number crunching, CAD programs, etc..), and sequential programming would be practical for extremely simple equations for execution.
5. Identify the basic form of data and task parallelism in computational problems
   1. Problem -> Instructions -> Processor
6. Explain the differences between processes and threads
   1. A process is a program that is in execution, while a thread is a part of the process, which can have multiple within one process (hence, multi-threading). Threads typically will take less time to terminate.
7. What is OpenMP and what is OpenMP pragmas?
   1. OpenMP is a library that supports multiprocessing in certain languages (i.e. C, C++, Fortran). OpenMP Pragmas is a compiler that that enable the compiler to generate threaded code.
8. What applications benefit from multi-core (list four)?
   1. 7Zip
   2. Android Studio
   3. Flight Simulators
   4. AutoCAD
9. Why Multicore? (why not single core, list four)
   1. Multicore will benefit applications that can have multiple processes running at once, most all developer/creator programs use multithreading.
   2. It uses less power than single core processors
   3. It improves clock speeds in computers
   4. There is less travel time between signals, which means that the processer will degrade less over time.

Tony Ngo - Project 2 Task 3B

A screenshot of a cell phone

Description automatically generated Picture 1: Code before fixA screenshot of a cell phone

Description automatically generatedPicture 2: Threads before fix

A screenshot of a cell phone

Description automatically generatedPicture 3: Code post-fix

A screenshot of a cell phone

Description automatically generatedPicture 4: Threads post-fix

I originally created the program using the nano text editor on the terminal. I copied the spmd2 program and saved and exited the program and made the executable using “***gcc spmd2.c -o -fopenmp***” command, which created an executable called smpd2, when I originally ran the program using the command “***./spmd2 4***”, the thread IDs repeated, which was not supposed to happen. So I went back into the code and changed lines 12/13 to initialize the ints within the function rather than outside. I remade my executable using the “***gcc spmd2.c -o spmd2 -fopenmp***” command and reran it using the command “***./spmd2 4***”, and I did not have repeating threads as seen from the fourth picture.