

Project\_A2

TASK3

Parallel Programming Skills:

* Identify the components on the Raspberry PI B+ (5)

CPU/RAM, Power socket, HDMI socket, ethernet controller, USB sockets, ethernet socket, camera connecting pin, display connecting pin, SD card slot, transistors, capacitors, diodes, serial buses, audio connector and many more.

* How many cores does the raspberry pi's B+ CPU have? (5)

The raspberry pi B+ version is a quad-core; so, four cores.

* List three main differences between x86(CISC) and ARM Raspberry PI(RISC). Justify your answer and use your own words (do not copy-paste) (8)

x86 processor has relatively fewer registers as compared to ARM. This is because x86 has a complex and ARM has reduced instruction sets.

x86 is used in PCs, large workstations and supercomputers whereas ARM is used for smaller purpose devices such as phones, routers, wearables, and tablets. Since ARM uses reduced instructions, it became easy to use them into smaller devices with fewer transistors and integrated circuitry.

ARM processor uses a register-register or load/store memory model but x86 uses the register-memory memory model. This is two separate ways of computing while in a clock cycle of instruction. ARM needs both operands in its register whereas for intel it will compute by loading one into its register and others can be done directly from the memory.

* What is the difference between sequential and parallel computation and identify the practical significance of each? (6)

The main difference between sequential and parallel computation is that in sequential only one instruction is handled by a processor at a time whereas in parallel computation multiple instructions are handled at a time using multiple cores/processors. Multi-tasking has been made easier due to this parallel computation technology. Editing a video while playing it is possible while using parallel computing but since only one instruction is processed at a given time both tasks cannot be possible in sequential computing.

* Identify the basic form of data and task parallelism in a computational problem. (5)

Data Parallelism:

One program 🡪 multiple data; single operation using multiple data while running a parallel computation is the base form of this problem-solving scheme. The data parallelism allows programmers to write scalable programs though the size of input and output.

Task/thread Parallelism:

The specification is given to multiple functions or threads rather than to data parallelism. In this system of parallelism, however, the computation is difficult to scale as it is hard to ensure that all works are done properly that contributed to the result.

* Explain the differences between process and threads. (6)

An executing instance of a thread is called a process.

The process is sure to execute but a thread may not always execute.

Thread is a subset of a process.

Usually, a process has only one thread of control – one set of machine instructions executing at a time.

All the threads running within a process share the same address space.

Usually, a process has only one thread of control – one set of machine instructions executing at a time.

* What is OpenMP and what are OpenMP pragmas? (3)

An OpenMP is a compiler with an implicit multithreading model reducing the error to a programmer. OpenMP pragmas in code mark through special directives that marks the programs into parallel computation models. Here each slave threads run independently completing its tasks and recombine later into its master while completing their tasks.

* What applications benefits from multi-core (List four)? (4)

Database server machines

Web servers

Scientific application

CAD/CAM

Multimedia applications

Compilers

* Why multi-core? (Why not single-core, list four?) (4)

Multi-core allows us to;

Process output faster

The need for multi-threaded tasks such as multi-tasking at a time in a computer

The computer does not get hot very often as all programs get executed faster

Low probability of signal degradation as it is processed in a short time.

The need for more parallelism in computing.