Michael Knight

Parallel Programming Skills:

Part: A

1. Identifying the components on the raspberry Pi B+.

Ethernet port, 4 USB ports, HDMI port, micro USB port, Ethernet controller, CPU and RAM, Display

1. How many cores does the Raspberry Pi’s CPU have?

Quadcore - 4

1. List three main differences between X86(CISC) and ARM Raspberry PI(RISC). Justify your answer and use your own words.

ARM uses less power, due to the fact that their cores don’t require heatsinks to help reduce their heat.

ARM is primarily used for programing of smaller devices where X86 is used for larger computer programing.

RISC is a more register based programming then CISC uses more memory.

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

Sequential computation is as the name suggests one task after another. Easier code.

Parallel computation helps reduce downtime while the CPU waits to receive information to carry on with the next task, it goes ahead and starts the next request so instead of waiting as long it starts the next step while it waits for the information.

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

Data Parallelism allows multiple data operations to be computated at once.

Task Parallelism refers to a running multiple tasks at once.

1. Explain the differences between processes and threads.

A process is when a program runs independently, and threads are multiple processes that share information.

1. What is OpenMP and what is OpenMP pragmas?

OpenMP is an interface which supports parallel computation, and pragmas is a complier that supports multithreaded code.

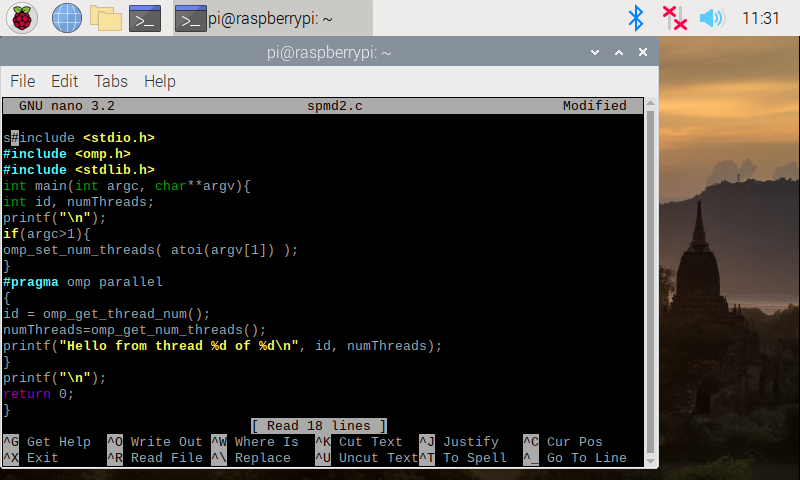
1. What applications benefit from multi-core(list four)?

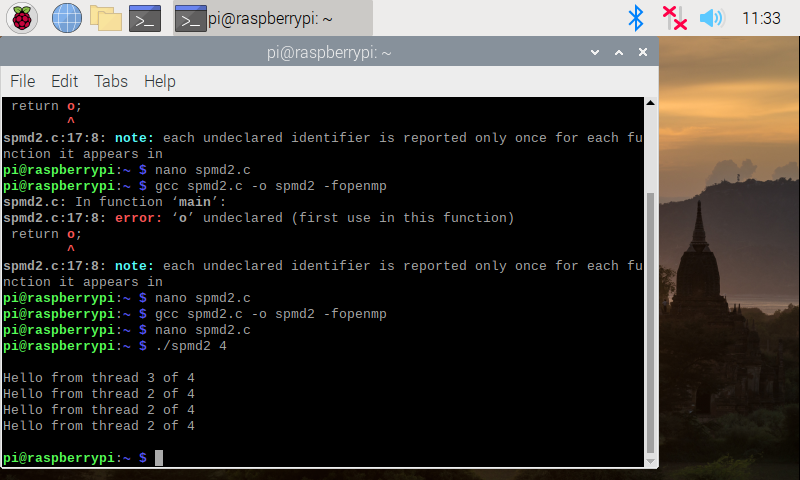
Multimedia applications, Web Servers, CAD/CAM, Compliers

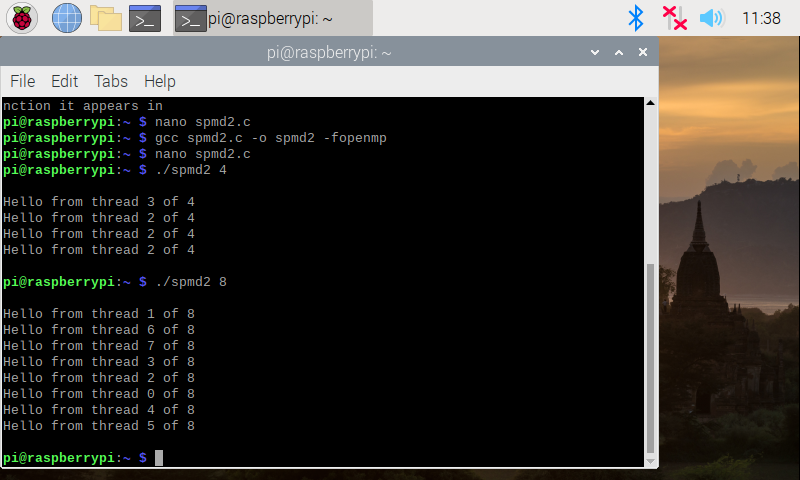
1. Why Multicore?(why not single core, list four)

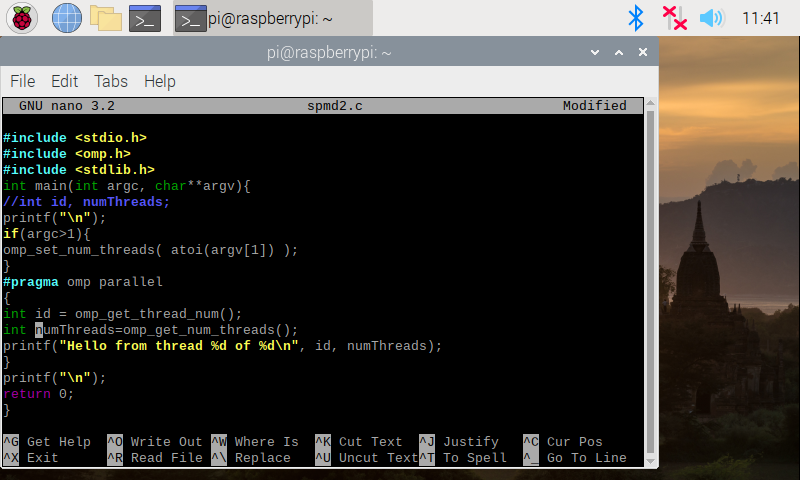
Faster results while computing, allows for multitasking, more energy sufficient, more compact.

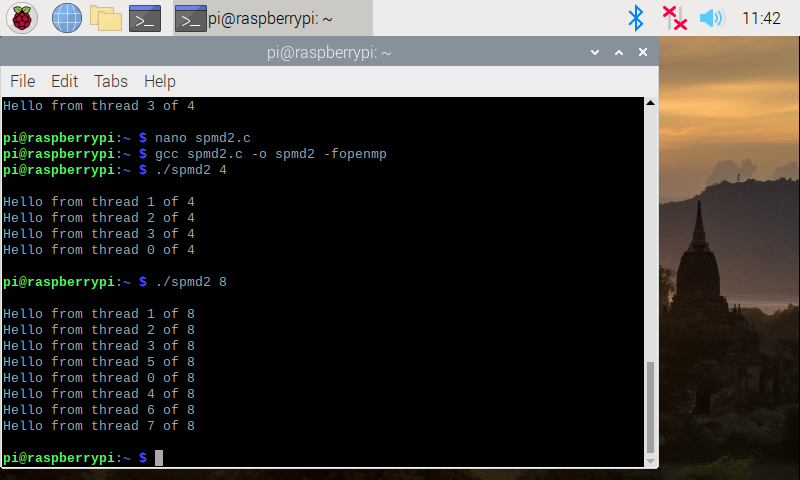
Part: B

First Using the code provided I ran this program.

Which gave me this result.

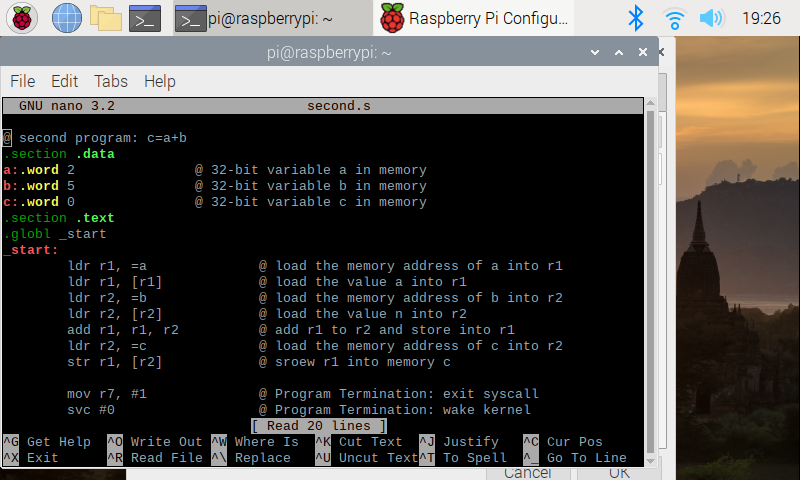
I would have thought each tread 1 threw 4 would be displayed but instead 3 2 2 2 was the result. I also ran tests. 

I then made the corrections to the code 

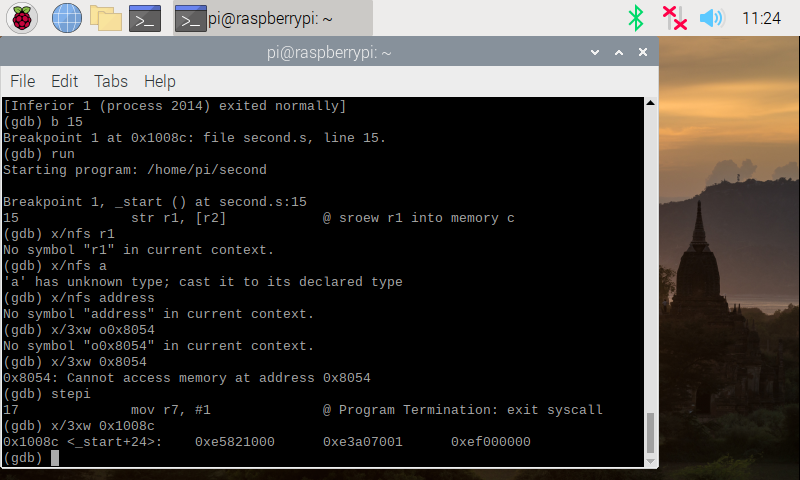
And ran the code. 

Arm Assembly Programming A2:

PartA:

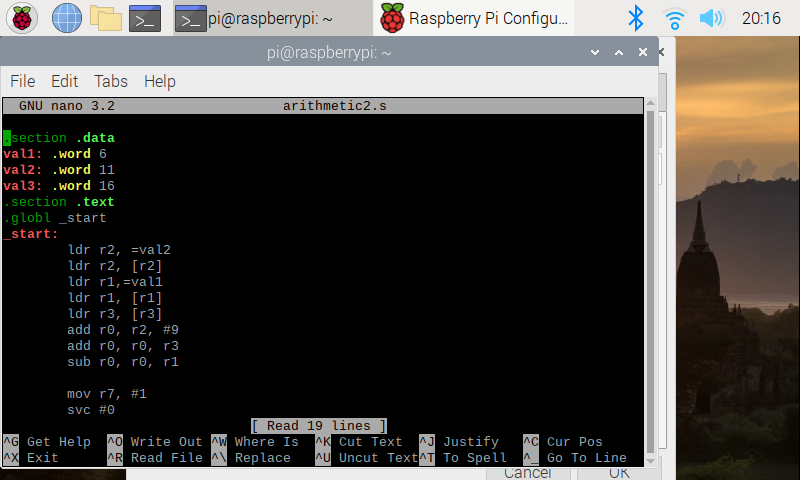
First I coded the sample code as instructed:

After assembling, linking, and running I saw no output because there is no output to display, its only storing and playing with registers.

Then ran debug on the program with a break point at 15 aswell as checking the register

Gave me the results 0x25821000 0xe3a07001 0xef000000

Part2:

Coded what was instructed Arithmetic2:

Then I ran Debug on it as instructed: