Hyoungjun Lee  
group name – gitter-geeks  
A3 Project  
computer org, programming  
3/5/2020

Parallel programming skills  
Foundation

Define the following: Task, Pipelining, Shared Memory, Communications, Synchronization. (in your own words)

**Task :** is an activity or works that you have to do, like part of project, in programming the task is a fundamental unit of programming that working frame work controls, operating system controls

**Pipelining:** process of accumulating and executing computer instructions, broken into several individual tasks by the processor.

**Communication:** how parallel data talk, exchange data each other

**Shared Memory:** part of random access memory, common memory location is shared between several processors through a bus, each processor has the same access to memory

**Synchronization:** process that coordinating two or more activities, devices or processes in same time.

**Classify parallel computersbased on Flynn's taxonomy. Briefly describe every one of them**

**Flynn’s taxonomy:** specific classification of parallel computer architectures based on number of concurrent instruction

(SISD) single instruction, single data : can only process and execute one instruction or data stream at a time

(MISD) multiple instruction, single data : parallel computer, excute more than one set of instruction, processing unit operates on the data independently via separate instruction using single data.

(SIMD) single instruction, multiple data : parallel computer, execute same instruction to various data thread at same time

(MIMD) multiple instruction, multiple data : parallel computer, every processor will be running into different instruction executes multiple instruction and data streams at the same time

**What are the Parallel Programming Model**

Shared memory, threads, distributed memory, data parallel, hybrid, SPMD, and MPMD

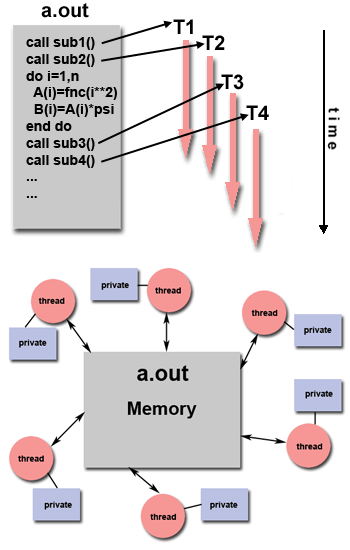
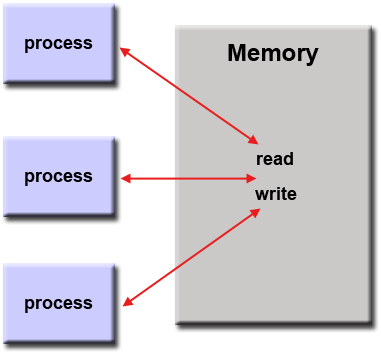
**List and briefly describe the types of Parallel Computer Memory Architectures. What type is used byOpenMP and why**

UMA(Uniform Memory Access) and NUMA (Non-Uniform Memory Access) , these are the types of parallel computer memory architectures, UMA have 1 shared general memory that can access multiple CPU, NUMA have several memories that connected to several CPU, OpenMP can be used both because it designed to multi processore, and shared memory machine.

Compare Shared Memory Model with Threads Model? (in your own words and show pictures

Shared memory model : memory is shared, each process can read and write to share address inside memory as shown in the picture, and this is basic type of memory model SMM is hard to manage data while implementing

Threaded Memory Model (**TMM**)

Program has a main body, single thread process can have multiple ways to execute, while implement, using OpenMP

Threaded memory

Shared memory

**What is Parallel Programming:**

A programming technique that is used to perform several computations or execution simultaneously, better option to complete task than serially

**What is system on chip(SoC)?**

This is a type of computer has the CPU, GPU, and RAM put onto a single chip,

**Does Raspberry PI use system on SoC?**

Yes

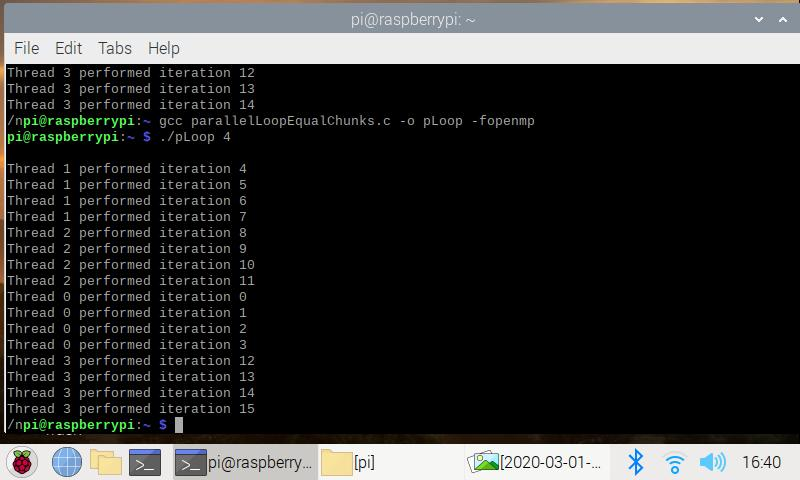
**Explain what the advantages are of having a System on a Chip rather than separate CPU, GPU and RAM components**

1 most advantage thing is reducing its size, refers it will be portable and less expensive, and better for smaller devices.

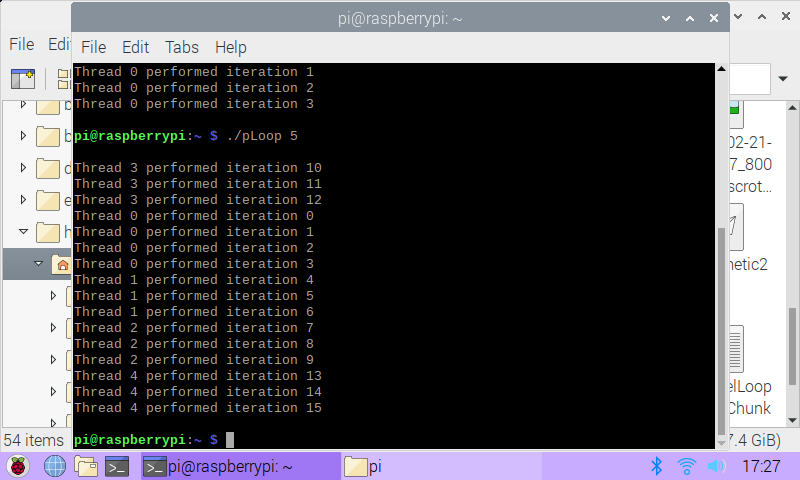
**40p) Parallel Programming Basics**

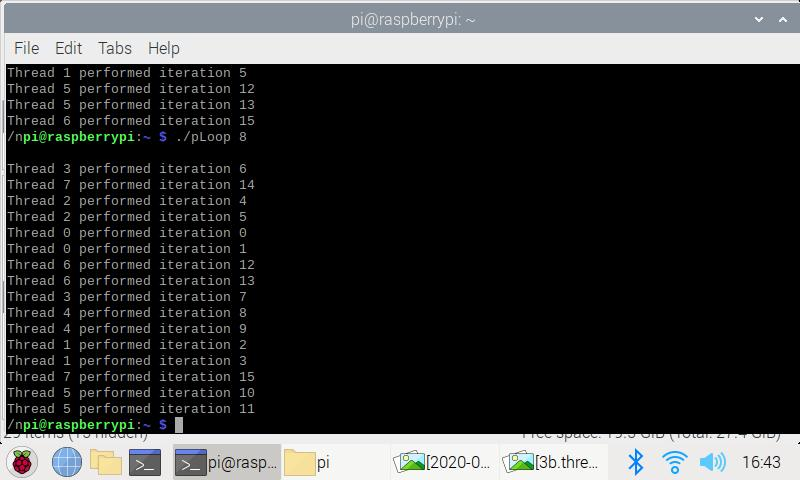
Firt pic

This is after I run the nano file, using ./pLoop and there is my output below



Also I tried more times to see how does pattern and output looks like

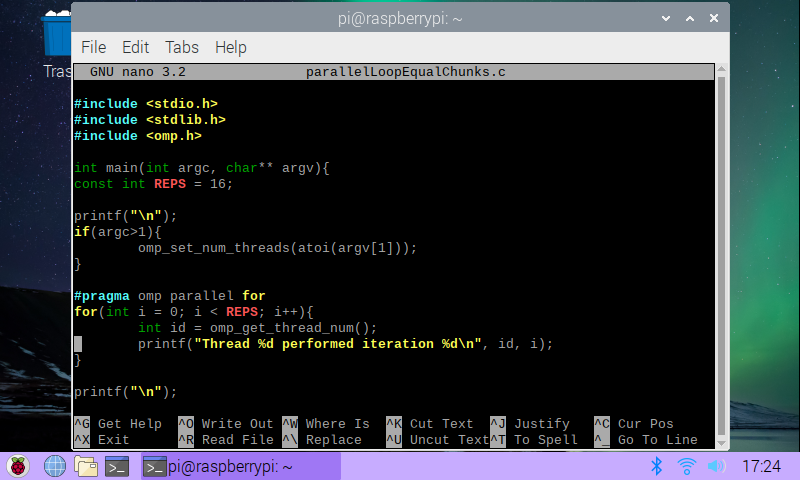




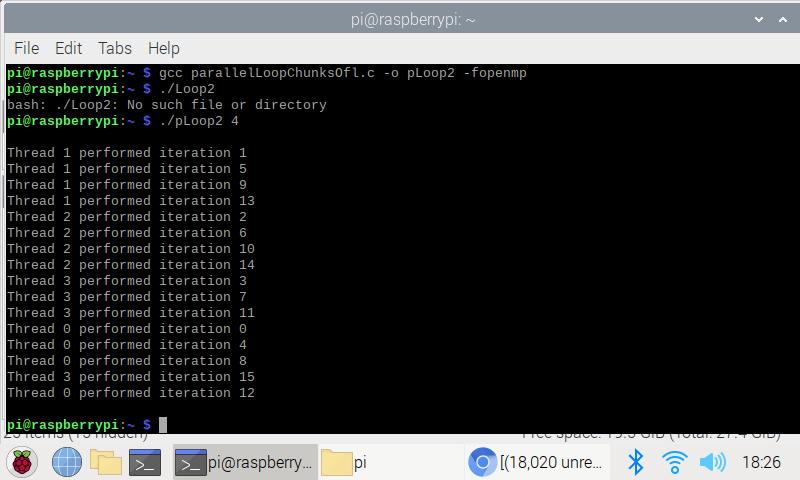
After watching some patterns, if you put number after ./pLoop, the thread will execute number between (0~number -1)

And there is an alternative way to divide an work,

I wrote parallelLoopChunksOf1.c



I ran the program using ./pLoop 4 and below is my output.



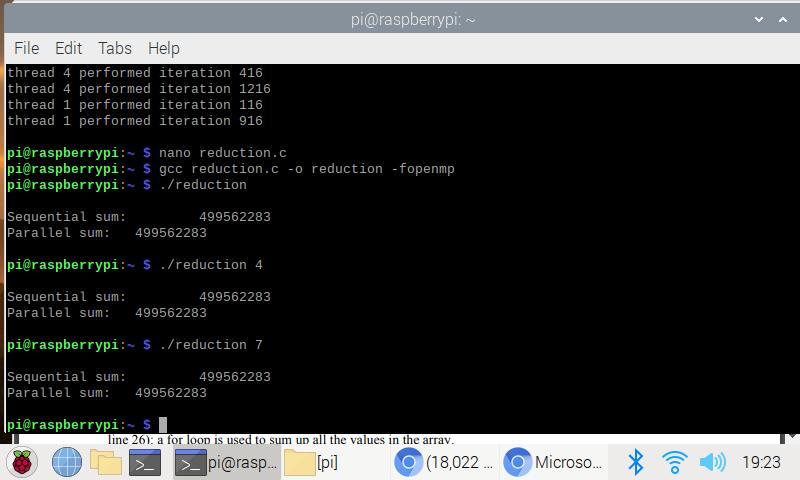
I think its not quite distributed evenly, and used static scheduling system.

Output is quite different compare to first one

This is like different order but same output

**3.4 Dynamic scheduling for time-varying task**

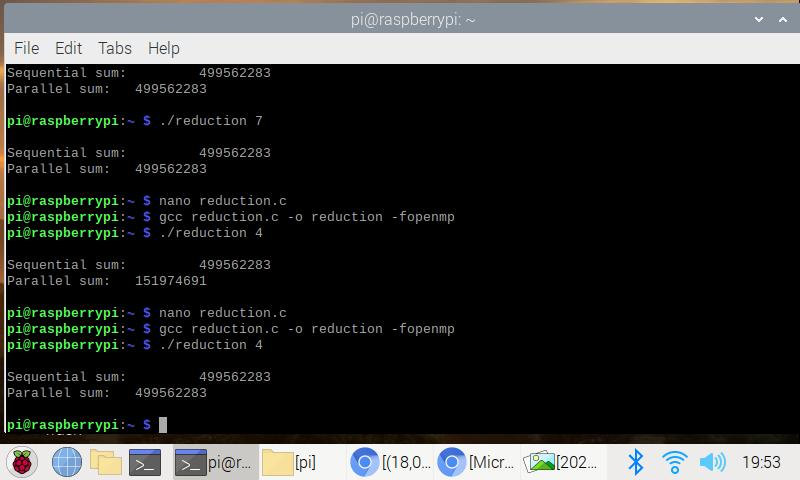
I wrote reduction.c by using nano program, and I compiled it and ran it,



I put iteration input as 4 first, both parallel sum and sequential sum are the same,

In order to see different output, we uncommented the first comment and see the result

I put both output for commented and uncommented



Output result

Sequential sum = 499562283

Parallel sum = 151974691