

Programación Paralela: Práctica de laboratorio 1

Lenin Pavón Álvarez ¹, Lucía Martínez Rivas², Miguel Ángel Perez de León³

¹Ayudante de Laboratorio, Facultad de Ciencias, UNAM

²Ayudante, Facultad de Ciencias, UNAM

³Profesor, Facultad de Ciencias, UNAM

Febrero 2024

Índice

1	Introducción	1
1.1	Requerimientos	2
1.2	Formato de entrega	2
2	Conceptos básicos	2
2.1	¿Qué es una supercomputadora?	2
2.2	El supercómputo en México	3
2.3	Conociendo mi equipo	4
3	El lenguaje del supercómputo	4
3.1	Python	4
3.2	Comandos en Bash	4
3.3	C	5

1. Introducción

Esta es la primera práctica de laboratorio del Seminario de Ciencias de la Computación A: *Introducción a la programación en paralelo con MPI, OpenMP y CUDA*. Está dividida en dos partes: el marco teórico y la parte práctica.

1.1. Requerimientos

Para realizar esta práctica se debe contar con acceso a internet y a un equipo de cómputo, y sea local o remoto. Para realizar los ejercicios del apartado 3.2 se debe contar con un sistema operativo Linux. Éste puede ser una máquina virtual local o remota (por ejemplo MACTI), en este último caso se debe contar con internet para realizar la conexión.

Para realizar los ejercicios del apartado 3.1 se requiere tener instalado Python 3 en el equipo de cómputo. Por último, para realizar los ejercicios del apartado 3.3 se requiere de un compilador de C instalado, gcc satisface este requisito.

1.2. Formato de entrega

Se deben de subir los siguientes archivos al repositorio de GitHub en un directorio un directorio con tu nombre y primer apellido (e.g. LeninPavon):

1. Un archivo de markdown (o en su defecto un pdf) respondiendo a las preguntas teóricas.
2. El código fuente correspondiente al apartado 3.1 en un archivo con extensión .py.
3. El código fuente correspondiente al apartado 3.2 en un archivo con extensión .sh.
4. El código fuente correspondiente al apartado 3.1 en un archivo con extensión .c.

Otro formato de entrega aceptable es

1. Un notebook (archivo con extensión .ipynb) donde se respondan las preguntas teóricas, y se encuentre el código fuente correspondiente al apartado 3.1
2. Los archivos fuentes correspondientes a los apartados 3.2 y 3.3.

2. Conceptos básicos

2.1. ¿Qué es una supercomputadora?

No suele haber una definición aceptada acerca de cuando se cruza la frontera del cómputo personal para pasar formalmente a ser supercómputo. Sin embargo, una definición de trabajo suele ser:

Si lo puedes hacer en tu computadora personal, no es supercómputo.

Y hay varios problemas prácticos (e.g. simulación del clima, predicción de terremotos, modelación de yacimientos petroleros) que no se pueden realizar en una computadora de escritorio. Es así que surge la necesidad de tener equipos que existan en la frontera de lo que es posible hacer computacionalmente.

Todos los años en la *Supercomputing Conference* se publica una lista ([enlace](#)) de las 500 supercomputadoras más poderosas (de las que se conoce su existencia públicamente), se les conoce como el TOP500.

Contesta las siguientes preguntas acerca del TOP500

1. ¿Qué significa la palabra Flop? ¿Qué es Rmax y Rpeak?
2. ¿Cuál es la supercomputadora que lidera el TOP500? ¿Cuáles son sus especificaciones?
3. ¿Cuáles son los 5 vendedores que lideran el TOP500?
4. ¿Cuáles son los países que dominan el TOP500? (Extra: grafica la tabla de países en el TOP500 usando Python)
5. ¿Cuál es la familia de sistema operativo que usan las computadoras del TOP500?

Hint: No se te olvide consultar la [página de estadísticas del TOP500](#)

2.2. El supercómputo en México

Aunque parezca que México existe en la periferia del desarrollo de supercómputo, existe una comunidad muy activa y con una gran cantidad de conocimientos tanto en la UNAM como en otras universidades. Ejemplo de esta creciente comunidad es la Delta Metropolitana que une clústers en la UAM, el IPN y la UNAM.

La DGTIC, a lo largo de su evolución desde el Programa Universitario de Cómputo, se ha convertido en el líder de supercómputo a nivel universidad. Anualmente se publica una convocatoria para que académicos e investigadores puedan solicitar tiempo de cómputo en la supercomputadora de la UNAM: Miztli.

Contesta las siguientes preguntas acerca del supercómputo en México

5. ¿Cómo se llamó la primera supercomputadora de la UNAM? ¿Cuáles eran sus especificaciones?
6. ¿Cuál son las especificaciones de Miztli? ¿Cuántos años tiene?

7. La Facultad de Ciencias de la UNAM tiene un Taller de Cómputo de Alto Rendimiento, que cuenta con equipo de cómputo para académicos de la facultad, ¿cuáles son sus especificaciones? (pregúntenle a Mike)

2.3. Conociendo mi equipo

Aunque la programación en paralelo permite aprovechar estos equipos a su máximo potencial (o al menos lo intenta), se puede usar programación en paralelo en varios contextos. Inclusive en una computadora personal, conociendo sus limitaciones, podemos acelerar considerablemente el rendimiento de nuestros programas. Pero para lograrlo primero necesitamos conocer nuestro equipo:

8. ¿Qué es la tabla de procesos en Linux? ¿Qué es un proceso?
9. ¿Cuál es la diferencia entre un hilo (thread) y un proceso?
10. ¿Cuántos procesadores (físicos y lógicos) tienes? ¿Cuál es la cantidad máxima de hilos que puede tener tu equipo de cómputo?

3. El lenguaje del supercómputo

3.1. Python

Recientemente se han hecho grandes avances para poder usar Python en contextos de cómputo de alto rendimiento. Al ser un lenguaje tan popular, tiene una gran cantidad de bibliotecas que pueden ser usadas en diferentes contextos.

Realiza el siguiente programa en Python

11. Escribe una función en Python que reciba como entrada dos números enteros: n (la cantidad de filas) y m (la cantidad de columnas). El programa debe de realizar una cuadrícula como la que se ve en la figura 1.

Sólo se requiere de ciclos (`for`), condicionales (`if`) y funciones (`def`).

3.2. Comandos en Bash

Ser flojo es una cualidad necesaria para tener éxito como programador. La automatización de procesos requiere de una alta capacidad de abstracción. Sin embargo, sólo se adquiere esta habilidad por medio del intento y error.

Aunque usualmente se usan los comandos uno por uno en la terminal, puedes crear un archivo donde en cada línea se encuentre un comando. Esto, de manera similar a un

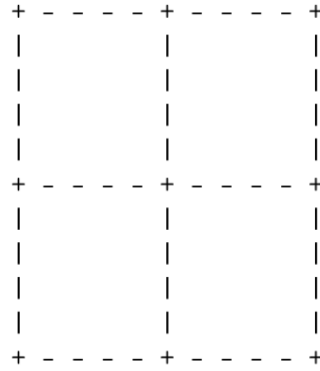


Figura 1: Ejemplo de una cuadrícula de 2x2

programa de Python u otro lenguaje de programación. Esto permite reutilizar el mismo código porque se le pueden dar entradas para adaptarlo a diferentes situaciones. A este tipo de archivo se les llama scripts y suelen tener la terminación `.sh`.

Escribe un script que realice las siguientes acciones

12. Imprime el mensaje “Iniciando programa”. Crea un directorio llamado Snoopy. Clona un repositorio (e.g. [Awesome made by mexicans](#)). Entra al directorio e imprime sus contenidos en el formato largo. Sal del directorio e imprime los contenidos. Imprime “Repositorio creado” y termina el programa.

Recuerda poner el shebang al inicio de tu programa

```
#!/bin/bash
```

3.3. C

Aunque C es un lenguaje con una gran historia, sigue teniendo utilidad de manera contemporánea. Se siguen actualizando y manteniendo sus compiladores, además de que existe una gran cantidad de infraestructura digital diseñada y optimizada para este lenguaje.

Escribe un programa en C que realice lo siguiente

13. Realiza una calculadora que sólo sume y reste. Adjunta capturas de las siguiente operaciones
 - a) $2 + 2$
 - b) $2 - 3$
 - c) $2.0 + 2$