Resolución de la Primera Practica Calificada de Programación Paralela

1. **Explique con sus palabras ¿Qué es un proceso de una computadora?**

   Es la ejecución de un grupo de instrucciones que se dan por parte del microprocesador dependiendo de lo que indique el programa que realiza el proceso.

2. **Explique a que se refiere cuando hablamos de una comunicación punto a punto entre 2 procesos, proponer un ejemplo en código.**

   Es el proceso de comunicación que se da mediante las siguientes funciones:

   -MPI_Send () : Se envia el mensaje hacia un destino.
   -MPI_Recv () : El receptor recibe el mensaje.

3. **¿Qué es una memoria RAM (principal), Cache y Virtual? E indicar ¿Cómo funcionan?**

   -Memoria RAM: En la RAM se guarda distinto tipo de información, desde los procesos temporales como modificaciones de archivos, hasta las instrucciones que posibilitan la ejecución de las aplicaciones que tenemos instaladas en nuestra PC.

   -Memoria Cache: El proceso que realiza la memoria caché es guardar las ubicaciones en el disco que ocupan los programas que han sido ejecutados, para que cuando vuelvan a ser iniciados el acceso a la aplicación logre ser más rápido.

   -Memoria Virtual: Memoria Virtual es el uso combinado de memoria RAM en su computadora y espacio temporero en el disco duro. Cuando la memoria RAM es baja, la memoria virtual mueve datos desde la memoria RAM a un espacio llamado archivo de paginación.

4. **¿En qué consiste la programación en Memoria Distribuida y la programación en Memoria Compartida?**

   -Memoria Distribuida: Cada proceso tiene su espacio de memoria, se puede realizar en diferentes memorias teniendo una memoria principal.
   -Memoria Compartida: Cada proceso tiene acceso a toda la memoria y comparten una única memoria.

5. **Describa en 3 lineas como máximo e indicar los parámetros de los siguientes comandos del MPI:**

-MPI_Send(): Realiza el envío de un mensaje de un proceso fuente a otro destino.

int `MPI_Send`(void *buf, int count, `MPI_Datatype` datatype,
`int dest, int tag, MPI_Comm comm)`

-MPI_Recv(): Rutina de recibimiento de un mensaje desde un proceso.

int **`MPI_Recv`**(void *buf, int count, **`MPI_Datatype`** datatype, int source, int tag, **`MPI_Comm`** comm, **`MPI_Status`** *status)

-MPI_Reduce(): Reduce un valor de un grupo de procesos en un único proceso raíz.

int **`MPI_Reduce`**(void *sendbuf, void *recvbuf, int count, **`MPI_Datatype`**
`datatype, MPI_Op op, int root,` **`MPI_Comm`** `comm)`

-MPI_AllReduce(): Reduce un valor de un grupo de procesos y lo redistribuye entre todos.

int **`MPI_Allreduce`**(void *sendbuf, void *recvbuf, int count,
`MPI_Datatype` `datatype, MPI_Op op,` **`MPI_Comm`** `comm)`