**TEORÍA**:

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

Es una serie o conjunto de instrucciones que hacen uso de un procesador.

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

Una comunicación punto a punto es el mecanismo básico de transferencia de mensajes entre un par de procesos uno, enviando y el otro, recibiendo.

MPI provee de un conjunto de funciones de envío y recepción de mensajes que permiten la comunicación de datos de cierto tipo con una etiqueta (tag) asociada.

Ejemplo:

**int MPI\_Send**(void \*buf, int count, MPI\_Datatype dtype, int dest, int tag, MPI\_Comm comm)

Envia un mensaje

**int MPI\_Recv**(void \*buf, int count, MPI\_Datatype dtype, int source, int tag, MPI\_Comm comm, MPI\_Status \*status)

1. ¿Qué es una memoria Ram (principal), caché y virtual? E indicar cómo funcionan.

Ram Principal: es la memoria principal en la cual se pueden realizar operaciones de lecturas y escrituras.

Memoria Caché: es la memoria volátil, su función es almacenar instrucciones y datos a los que el procesador debe acceder continuamente.

Memoria Virtual: reserva un espacio en el disco duro, que simule la memoria ram.

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

La programación en memoria distribuida se refiere a cuando cada proceso hace uso de su propia memoria, en cambio, en memoria compartida, varios procesos pueden hacer uso de la misma sección de memoria.

1. Describa en 3 lineas como máximo e indicar los parámetros de los siguientes comandos del MPI:
2. int MPI\_Send(void \*buf, int count, MPI\_Datatype dtype, int dest, int tag, MPI\_Comm comm)
3. int MPI\_Recv(void \*buf, int count, MPI\_Datatype dtype, int source, int tag, MPI\_Comm comm, MPI\_Status \*status)
4. int MPI\_Reduce(void \*bufsend, void \*bufrecv, int count, MPI\_Datatype dtype, MPI\_op operation, int dest, MPI\_Comm comm)

MPI\_Reduce combina los elementos provisto en el buffer de entrada (inbuf) de cada proceso en el grupo, usando la operación op (puede ser una operación predefinida o definida por el usuario) y retorna el valor combinado en el buffer de salida (outbuf) del proceso root.

El proceso root enviará a cada proceso un trozo del contenido del buffer de envío, comenzando desde de la dirección inicial de dicho buffer se desplazara una cantidad (incount) para realizar el siguiente envio

1. MPI\_Allreduce(inbuf, outbuf, count, type, op, root, comm)