RESOLUCIÓN DE PRACTICA DE PROGRAMACIÓN PARALELA

**PARTE TEORIA**

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

Un proceso es un conjunto formado por las instrucciones de un programa destinadas a ser ejecutadas por el microprocesador. Los procesos son creados y destruidos por el sistema operativo, El mecanismo por el cual un proceso crea otro proceso se denomina bifurcación (*fork*). Los nuevos procesos son independientes y no comparten memoria con el proceso que los ha creado.

2) Explique a que se refieren cuando hablamos de una comunicación punto a punto entre 2 procesos , proponer un ejemplo

se refieren a una conexión limitada a dos extremos o nodos, en la cual uno envía datos y el otro los recibe.

#include "mpi.h"

#include <iostream>

using namespace std;

int main(int argc, char \*argv[])

{

int rank, size, contador;

MPI\_Status estado;

MPI\_Init(&argc, &argv);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

if(rank == 0){

MPI\_Send(&rank ,1 ,MPI\_INT ,rank+1 ,0,MPI\_COMM\_WORLD);

}else{

MPI\_Recv(&contador ,1 ,MPI\_INT ,rank-1,0 ,MPI\_COMM\_WORLD,&estado);

cout<<"Soy el proceso "<<rank<<" y he recibido "<<contador<<endl;

contador++;

if(rank != size-1)

MPI\_Send(&contador, 1 ,MPI\_INT ,rank+1 , 0 ,MPI\_COMM\_WORLD);

}

MPI\_Finalize();

return 0;

}

3) ¿Que es una memoria Ram (principal), Cache y Virtual? E indicar ¿Como funcionan?

RAM: La memoria RAM es la memoria principal de un dispositivo donde se almacena programas y datos informativos. La memoria RAM es conocida como memoria volátil lo cual quiere decir que los datos no se guardan de manera permanente, es por ello, que cuando deja de existir una fuente de energía en el dispositivo la información se pierde.

Cache: Es muy pequeña y está incluida en el interior del micro. Su función es sencilla, conseguir que los datos más usados estén lo más cerca del procesador para ser accedidos de la manera más rápida posible.

MV : La memoria virtual permite simular una memoria RAM de mayor tamaño que la que tienes instalada en tu equipo. Es un mecanismo del cual se encarga el sistema operativo.

Su utilidad reside en que las aplicaciones y sus datos, para poder ser usados por el procesador, tienen que pasar del disco duro, que es donde se encuentra almacenado todo cuando apagas al equipo, a la memoria RAM que puede ser accedida de forma directa por el micro.

4) ¿En que consiste la programación en memoria distribuida y la programación en memoria compartida?

Memoria Distribuida: En esta arquitectura, el computador paralelo es esencialmente una

colección de procesadores secuenciales, cada uno con su propia

memoria local, que pueden trabajar conjuntamente.

Memoria Compartida: Existen dos tipos: Uniform memory access y Non-uniform memory access

. Uniform memory access (UMA) : Cada procesador tiene acceso uniforme a memoria. También se llaman symmetric multiprocessors (SMPs)

. Non-uniform memory access (NUMA) : El tiempo de acceso depende de dónde están los datos. El acceso local es más rápido. Más fácil y barato de escalar que SMPs

5) Describa en 3 lineas como maximo e indicar los parametros de los siguientes comandos del MPI:

a) MPI\_Send(…) : Realiza una operación de envío en modo estándar y regresa cuando el búfer de envío se puede reutilizar de forma segura.

int MPI\_Send(

void \*data,

int count,

MPI\_Datatype datatype,

int dest,

int tag,

MPI\_Comm communicator);

b) MPI\_Recv(…): Realiza una operación de recepción y no regresa hasta que se reciba un mensaje coincidente.

Int MPI\_Recv(

void \*data,

int count,

MPI\_Datatype datatype,

int source,

int tag,

MPI\_Comm communicator,

MPI\_Status status);

c) MPI\_Reduce(…) Realiza una operación de reducción global en todos los miembros de un grupo. Puede especificar una operación matemática o lógica predefinida o una operación definida por la aplicación.

int MPI\_Reduce(

void \*input\_data,

void \*output\_data,

int count,

MPI\_Datatype datatype,

MPI\_Operator Op,

int tag,

MPI\_Comm communicator);

d)MPI\_Allreduce(…): Combina valores de todos los procesos y distribuye el resultado a todos los procesos.

int MPI\_Allreduce(

void \*sendbuff,

void \*recvbuff,

int count,

MPI\_Datatype datatype,

MPI\_Operator Op,

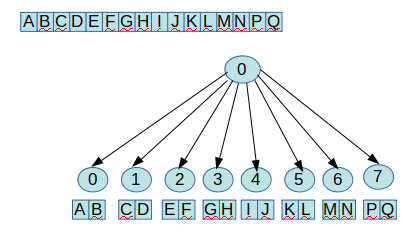
int tag,

MPI\_Comm communicator);

**PARTE PRACTICA**

**5. Suponga que comm\_sz=8 y la cantidad de elementos es n=16**

**a) Diseñe un diagrama que explique como MPI\_Scatter puede ser implementado usando comunicaciones basadas en arboles. Puede suponer que el origen del scatter es el proceso con rank.**



**b) Hacer lo mismo para el MPI\_Gather, en este caso con el proceso 0 como destino.**

