# Solución de la Práctica Calificada # 1 de Prog. Paralela

## TEORÍA

1) ***Explique con sus palabras ¿Qué es un proceso de una computadora?***
Se refiere a la ejecución de diversas instrucciones por parte del microprocesador, de acuerdo a lo que indica un programa.

2) ***Explique a que se refieren cuando hablamos de una comunicación punto a punto entre 2 procesos, proponer un ejemplo en código.***
Trata sobre comunicación de 2 procesos, donde uno se encarga de enviar mensajes y el otro los recibe, o viceversa, utiliza las funciones de Send y Recv de MPI.

Ejemplo:

```
#include "mpi.h"
#include <iostream>
using namespace std;

int main(int argc, char *argv[]) {
int rank, size;
char msje[]="Hola";

MPI::Init(argc, argv);
MPI_Status status;
size = MPI::COMM_WORLD.Get_size();
rank = MPI::COMM_WORLD.Get_rank();

if(rank==0){
        MPI_Send(&msje,4,MPI_CHAR,1,0,MPI_COMM_WORLD);
        cout<<"Estoy enviando el mensaje al Proceso 1: "<<msje<<endl;
}

if(rank==1){
        MPI_Recv(&msje,4,MPI_CHAR,0,0,MPI_COMM_WORLD,&status);
        cout<<"Estoy recibiendo el mensaje de Proceso 0: "<<msje;
}

MPI::Finalize();
return 0;


}
```

3) ***¿Qué es una memoria RAM (principal), Cache y Virtual? E indicar ¿Cómo funciona?***
**RAM:** Es un tipo de memoria operativa de los computadores y sistemas informáticos, adonde va a ejecutarse la mayor parte del software: el propio sistema operativo, el software de aplicación y otros programas semejantes.

Aloja las utilidades y datos con los que trabajas en un determinado momento. Si no existiera, el procesador se aburriría esperando a que el disco duro le mandara algo.

**CACHE:** Sirve para almacenar temporalmente los datos recientemente procesados

Cuando accedemos a un dato cualquiera en nuestro sistema computarizado, se crea de inmediato una copia de los datos más relevantes del mismo en la memoria caché, de modo que los accesos siguientes a dicha información la tengan a mano y no deban rastrearla hacia su lugar de origen

**VIRTUAL:** Permite simular una memoria principal de mayor tamaño cuando la capacidad de la Ram ya se encuentra repleta.

Cuando la memoria RAM es baja, la memoria virtual mueve datos desde la memoria RAM a un espacio del disco duro, memoria virutal y su velocidad es mucho mas lenta que el de la memoria principal.

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

La programación en la memoria compartida: Donde los núcleos están conectados a una sola memoria principal, cada uno de ellos puede tener acceso a cualquier parte de la RAM.

La programación en memoria distribuida: Donde los núcleos están conectados a una red, y cada uno de ellos tiene acceso directo a su propia memoria.

5) ***Describa en 3 líneas como máximo e indicar los parámetros de los siguientes comandos en MPI:***

a) **MPI_Send():** Función que se encarga del envío de mensajes hacia otro(s) procesador(es).

```
MPI_Send(
    Buffer de envío , //mensaje de envio
    Tamaño del mensaje ,
    Tipo del mensaje (Entero, Doble,etc) ,
    Proceso de Destino , //Hacia que proceso sera enviado el mensaje
    Etiqueta , //identificador que tendrá el proceso que recibe
    Comunicador // Protocolo de comunicación)
```

b) **MPI_Recv():** Función que se encarga de la recepción de mensaje desde otro proceso.

```
MPI_Recv(
    Buffer de llegada , //mensaje a recibir
    Tamaño del mensaje ,
    Tipo del mensaje, //(Entero, Doble,etc) ,
    Proceso de Origen , //Desde que proceso es enviado el mensaje
    Etiqueta , //identificador que tendrá el proceso que envia
    Comunicador , //Protocolo de comunicación
    Estado )
```

c) **MPI_Reduce():** Función que se encarga de realizar operaciones con todos los procesos, el resultado de esa operación es enviado a solo un proceso.

```
MPI_Reduce(
    Input , //indica el dato de entrada que sera evaluado en la operación asignada
    Output, //indica el dato de salida, como resultado de la operación asignada
    Tamaño del mensaje (output) ,
    Tipo del mensaje , //(Entero, Doble, etc)
    Tipo de operación , //Aqui se indica que operación realizaran todos los inputs
    Proceso de destino , //Que proceso almacenara el output luego de realizar la operación
    Comunicador //Protocolo de comunicación)
```

d) **MPI_Allreduce():** Función que se encarga de realizar operaciones con todos los procesos, el resultado de esa operación es enviado a todos los procesos.

```
MPI_AllReduce(
    Input , //indica el dato de entrada que sera evaluado en la operación asignada
    Output, //indica el dato de salida, como resultado de la operación asignada
    Tamaño del mensaje (output) ,
    Tipo del mensaje , //(Entero, Doble, etc)
    Tipo de operación , //Aqui se indica que operación realizaran todos los inputs
    Comunicador //Protocolo de comunicación)
```

## PRÁCTICA

***6) Utilizando MPI, implemente un algoritmo que determine el número de veces que un elemento x aparezca en un vector A con n elementos enteros. Se puede asumir que su algoritmo comienza con los elementos ya distribuidos entre los p procesos (n/p para cada uno).***

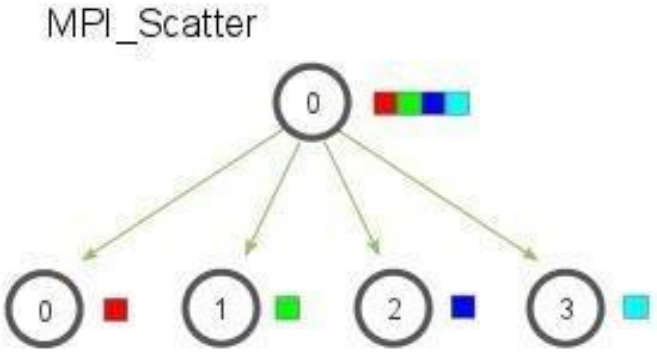
*(Dentro del repositorio Ej6)*

***7) Desarrolle un algoritmo MPI, utilizando p procesadores para calcular n!***

*(Dentro del repositorio Ej7)*

***8) Suponga que comm_sz=8 y la cantidad de elementos de 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 0.***




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