

Resolución de la 1ra. Práctica Calificada del curso de Programación Paralela

Teoría

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

Es la ejecución de un programa que posee una secuencia de instrucciones, un estado actual y un conjunto de recursos asociados.

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

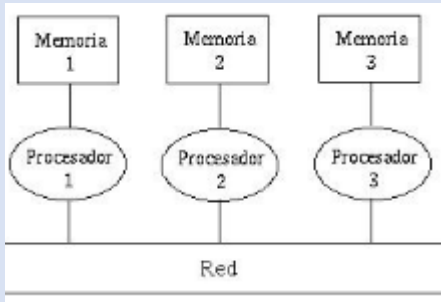
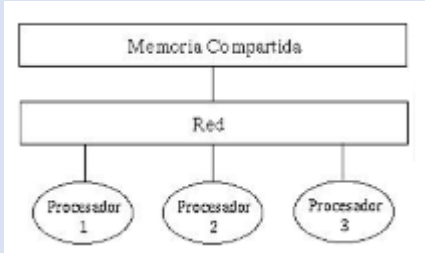
Es la comunicación que se da entre 2 procesos, el envío de mensaje el cual lo hace el MPI_Send () y el receptor el cual es el MPI_Recv (),
Veamos un código:

```
...
If (my_rank != 0){
    sprintf (msg, "Saludos del proceso %d de %d!", my_rank, comm_sz);
    MPI_Send(msg, strlen(msg) + 1, MPI_CHAR, 0, 0, MPI_COMM_WORLD);
} else {
    printf ("Saludos del proceso %d de %d!\n", my_rank, comm_sz ");
    for ( int q=1; q < comm_sz; q++) {
        MPI_Recv( msg, MAX_STRING, MPI_CHAR, q, 0, MPI_COMM_WORLD,
        MPI_STATUS_IGNORE);
        `printf ("%s\n", msg);
    }
}
....
```

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

Memoria RAM	Cache	Memoria Virtual
<p>Memoria RAM (Memoria de acceso aleatorio). Es una colección de posiciones. Es una memoria volátil.</p> <p>Función: Almacena información y luego es enviado las instrucciones y/o datos al procesador.</p>	<p>También llamada RAM caché Es parte de la memoria RAM estática de alta velocidad(SRAM)</p> <p>Función: Se encarga de registrar las instrucciones comúnmente ejecutadas de una aplicación.</p>	<p>Es una simulación de la memoria RAM. Se utiliza el principio de Localidad, swap- space y páginas.</p> <p>Función: Reserva un espacio en el disco duro para poder ser usados en el procesador.</p>

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

Programación en Memoria Distribuida	Programación en Memoria Compartida
<p>Cada proceso tiene su propio espacio de memoria. Los procesadores pueden compartir información solamente enviando mensajes.</p> 	<p>Todos los procesos comparten una única memoria. Cada proceso tiene acceso a toda la memoria.</p> 

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

- a) MPI_Send () : Es aquel que envía mensajes a un proceso con parámetros

MPI_Send (
void*	msg_buf	REFERENCIA
Int	msg_size	CANTIDAD
MPI_Datatype	datatype	TIPO DE DATO
Int	dest	RANK DESTINO
Int	tag	ETIQUETA
MPI_Comm)	communicator	COMUNICADOR

- b) MPI_Recv (): Es aquel que capta el mensaje del proceso

MPI_Recv (
void*	msg_buf	REFERENCIA DATO
Int	buf_size	CANTIDAD
MPI_Datatype	buf_type	TIPO DE DATO
Int	source	RANK DESTINO
Int	tag	ETIQUETA
MPI_Comm	communicator	COMUNICADOR
MPI_Status)	status	ESTADO

- c) MPI_Reduce (): Toma una matriz de elementos de entrada en cada proceso y devuelve una matriz de salida.

MPI_Reduce (
void*	input_data	REFERENCIA DATO
void*	output_data	REREFERENCI LLEG
int	count	CANTIDAD / DATOS
MPI_Datatype	buf_type	TIPO DE MENSAJE
MPI_Op	operator	OPERACION
Int	dest_proc	DESTINO / PROCESO
MPI_Comm)	comm	COMUNICADOR

- d) MPI_Allreduce (): Todos los procesos necesitan una reducción para poder continuar.

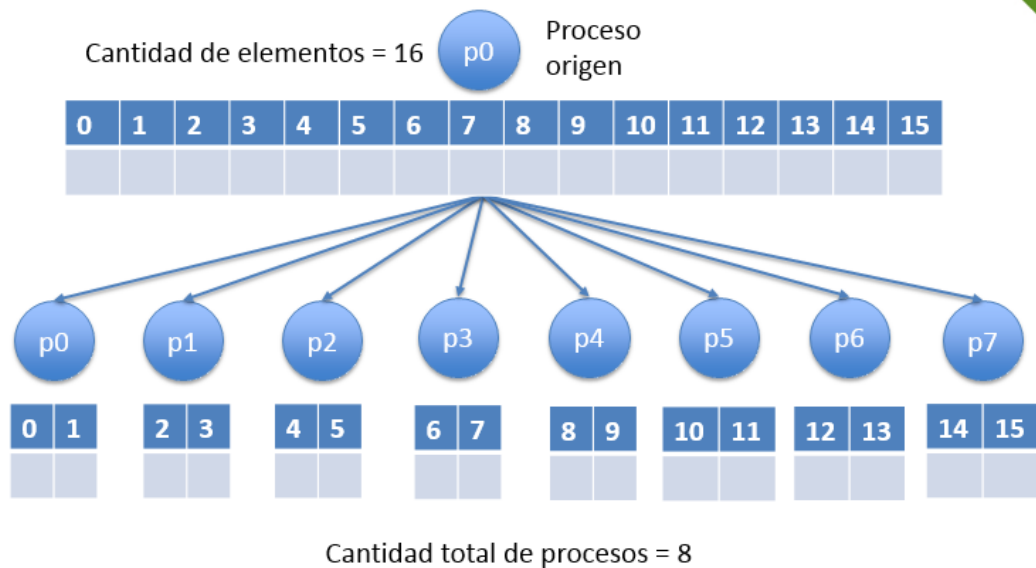
MPI_Allreduce (
void*	input_data	REFERENCIA DATO
void*	output_data	REFERENCIA LLEG
int	count	CANTIDAD / DATOS
MPI_Datatype	buf_type	TIPO DE MENSAJE
MPI_Op	operator	OPERACION
MPI_Comm)	comm	COMUNICADOR

Práctica

8) Suponga que $\text{comm_sz}=8$ y la cantidad de elementos es $n=16$

- a) Diseñe un diagrama que explique cómo MPI_Scatter puede ser implementado usando comunicaciones basadas en árboles. Puede suponer que el origen del Scatter es el proceso con Rank 0.

MPI_Scatter (): Toma una lista y lo distribuye equitativamente a cada proceso incluyéndose



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

MPI_Gather (): Entrega el valor y lo concatena todos los elementos de los procesos, lo recibe en orden mientras lo une.

