

Practica 2

Jose Ignacio Moreno Montesinos y Diego González Sanz

Ejercicio 1: La comunicación punto a punto es un medio de transmisión o comunicación con exactamente dos puntos finales, y no hay datos o paquetes de formato. El centro de computadoras en cada extremo tenía que asumir la plena responsabilidad para el formato de los datos transmitidos entre ellos. Además una conexión punto a punto sirve para llevar los datos del sistema local a un sistema remoto o bien de una red local a una red remota.

La siguiente foto son todos los datos primitivos de MPI. Además los tipos de datos primitivos son contiguos.

MPI_CHAR	MPI_C_COMPLEX
MPI_WCHAR	MPI_C_FLOAT_COMPLEX
MPI_SHORT	MPI_C_DOUBLE_COMPLEX
MPI_INT	MPI_C_LONG_DOUBLE_COMPLEX
MPI_LONG	MPI_C_BOOL
MPI_LONG_LONG_INT	MPI_LOGICAL
MPI_LONG_LONG	MPI_C_LONG_DOUBLE_COMPLEX
MPI_SIGNED_CHAR	MPI_INT8_T
MPI_UNSIGNED_CHAR	MPI_INT16_T
MPI_UNSIGNED_SHORT	MPI_INT32_T
MPI_UNSIGNED_LONG	MPI_INT64_T
MPI_UNSIGNED	MPI_UINT8_T
MPI_FLOAT	MPI_UINT16_T
MPI_DOUBLE	MPI_UINT32_T
MPI_LONG_DOUBLE	MPI_UINT64_T
	MPI_BYTE
	MPI_PACKED

Las ventajas de la comunicación punto a punto es que son fáciles de configurar, tiene una menor complejidad y el coste es inferior ya que no necesita dispositivos de red ni servidores dedicados.

Las desventajas de la comunicación punto a punto es que la administración es no centralizada, no son muy seguras, el rendimiento se reduce, no son escalables y por último el funcionamiento se puede ralentizar porque todos los dispositivos pueden actuar como cliente y como servidor.

Ejercicio 2:

```
diego@diego-OMEN-by-HP-Desktop-PC-880-p0xx:~/Arquitectura de computadores/Practica2$ mpirun -np 4 ./ejecutable2
Soy el proceso 0 y he generado el numero: 6
Soy el proceso 1 y he inicializado la variable con el dato: 14
Soy el proceso 2 y he inicializado la variable con el dato: 8
Soy el proceso 3 y he inicializado la variable con el dato: 12
Soy el proceso 0 y he recibido el numero: 6
Soy el proceso 0 y he recibido el numero: 14
Soy el proceso 0 y he recibido el numero: 8
Soy el proceso 0 y he recibido el numero: 12
Soy el proceso 0 y ya he recibido los datos de los 4 procesos: 6, 14, 8, 12, diego@diego-OMEN-by-HP-Desktop-PC-8
```

Primero debemos inicializar las variables rank, count, size, status, número y numerosAleatorios.

Después hacemos un bucle for para meter ahí los valores aleatorios que después lo enviaremos al proceso 0. Después si el rank es igual a 0 nos metemos en él if y primero mandamos al proceso 0 el número generado en la posición 0 después recibe ese número el proceso 0. Ahora recibo el número generado por los demás procesos y lo hacemos con un bucle for. Si es distinto a 0 el rank nos vamos al else, que aquí enviamos el número generado en la posición de rank al proceso 0. Para finalizar decimos si el rank es 0 que nos muestre los números recibidos de los otros procesos por pantalla, y después de esa condición la función MPI_Finalize();

Ejercicio 3:

En el ejercicio 3 hemos hecho un código en el que nos pedían realizar tantas vueltas de manera automática el enviar un mensaje de proceso a proceso. Hemos utilizado un mensaje el cual transmitimos. Luego hemos usado una señal de activo para saber el estado del proceso. Utilizamos tanto el send como receiver para controlar los procesos. Luego en función del rango en el que se encuentre el proceso mandara una cosa u otra y se mostrarán cosas por pantalla