

Tema 2-Modelo de paso de mensajes

El modelo de paso de mensajes consiste en comunicar uno o más procesos mediante rutinas que permiten recibir y enviar mensajes entre procesos. Es el programador el que tiene que evitar las dependencias, interbloqueos y race conditions.

El modelo de ejecución de un programa de paso de mensajes es un programa paralelo compuesto por múltiples tareas que utilizan su propia memoria local. Normalmente la relación es de una tarea por cada elemento de procesado. Cada envío de mensaje necesita un receptor. Pueden ser:

- **MPMD** (Multiple Program Multiple Data): cada tarea realiza su propio programa.
- **SPMD** (Single Program Multiple Data): todas las tareas comparten el mismo programa; se divide internamente el código.

La librería que se usa en C es `mpi.h`. Sus funciones tienen la sintaxis `MPI_nombre(parametros)` y devuelven `MPI_SUCCESS` o `MPI_ERR_*`. Los procesos son independientes hasta inicializarlos con `MPI_Init`, sincronizándose para poder intercambiar datos. Los procesos deben conocer el número de procesos, que se obtiene con `MPI_Comm_size` y su identificador, con `MPI_Comm_rank`. Al final del programa, con `MPI_Finalize`, se liberan todos los recursos reservados por MPI. El comunicador global que une todos los procesos se denota como `MPI_COMM_WORLD`.

La estructura básica de un programa MPI es:

```
#include <mpi.h>
#include <stdio.h>

int main(int argc, char *argv[]){
    int numprocs, rank, namelen;
    char processor_name[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    //código

    MPI_Finalize();
}
```

Tipos de operaciones

- **Punto a punto:** van de un proceso origen a un proceso destino. Tienen argumentos *dest* o *source*.
- **Colectivas:** involucran a todos los procesos a la vez. Suelen tener un argumento *root*.
- **Bloqueantes:** los procesos esperan a que el mensaje se reciba.
- **No bloqueantes:** los procesos continúan independientemente del receptor.

Tipos de datos

- MPI_INT
- MPI_LONG
- MPI_CHAR
- MPI_FLOAT
- MPI_DOUBLE
- MPI_SHORT

Tag y Status

El tag es un número que sirve para identificar mensajes punto a punto. Si no necesitamos ninguna etiqueta podemos usar la constante *MPI_ANY_TAG*.

El status es una estructura que contiene la información de la operación. Si no necesitamos el estado, podemos usar la constante *MPI_STATUS_IGNORE*. Contiene el número de elementos enviados y el origen y tag del envío.