



# Computación de Altas Prestaciones

Modelo de programación basado en el paso de mensajes

**José Luis Risco Martín**

Dpto. Arquitectura de Computadores y Automática  
Universidad Complutense de Madrid

This work is derivative of “Modelo de programación basado en el paso de mensajes”

by [Ignacio Martín Llorente](#), licensed under [CC BY-SA 4.0](#)



# Índice

1. Ventajas e inconvenientes
2. Características de los mensajes
3. Comunicaciones punto a punto
4. Comunicaciones colectivas

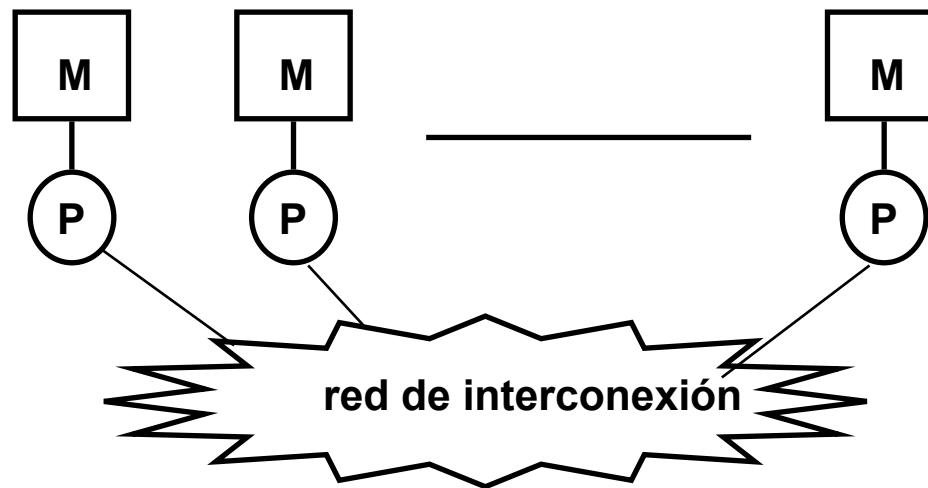


## Bibliografía:

- Introduction to High Performance Computing for Scientists and Engineers. Georg Hager and Gerhard Wellein. CRC Press. 2011 (Chapters 9 and 10, focused on MPI).

# Introducción

- Consiste en la replicación del paradigma de programación secuencial
- El programador divide la aplicación en varios procesos que se ejecutan en diferentes procesadores sin compartir memoria y comunicándose por medio de mensajes
- **Visión del programador:** Lenguaje secuencial con variables privadas + Rutinas de paso de mensajes



*“Lenguaje ensamblador de la programación paralela”  
Mejores eficiencias que con el resto de lenguajes*



# Ventajas e Inconvenientes

## ■ Ventajas:

- Portable de modo eficiente a cualquier tipo de arquitectura: computador paralelo, red de estaciones y a una única estación de trabajo

## ■ Inconvenientes:

- Bastante más complicado de programar y depurar que memoria compartida
- La eficiencia depende del programador

## ■ SPMD vs. MPMD

- Muchos computadores actuales solo soportan SPMD, el mismo ejecutable en cada procesador: IBM SP2 y CRAY T3E
- No es problema:

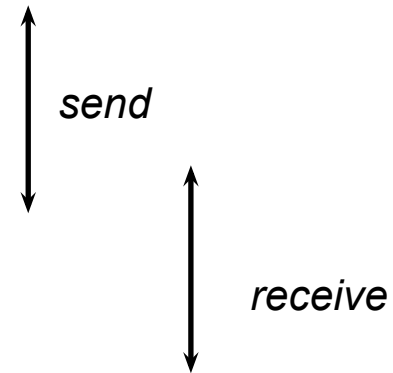
```
main(int argc, char **argv)
    if proceso es el controlador
        control(argumentos)
    else
        worker(argumentos)
```

- Muchos sistemas actuales no soportan tiempo compartido, los procesadores están siendo empleados por un único usuario (cola de trabajos en cada partición): IBM SP2 y CRAY T3E
  - El tiempo compartido se encuentra en investigación

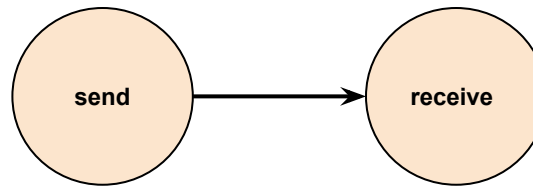


# Características de los Mensajes

- Un mensaje es una transferencia de datos de un proceso a otro proceso
- **Información que caracteriza un mensaje:**
  - En qué variable están los datos que se envían
  - Cuantos datos se envían
  - Qué proceso recibe el mensaje
  - Cuál es el tipo de dato que se envía
  - Qué proceso envía el mensaje
  - Donde almacenar los datos que se reciben
  - Cuantos datos espera recibir el proceso receptor
- **Partes de un proceso que se comunica por medio de mensajes:**
  - 1. Acceso: El proceso se debe enrolar en el sistema de paso de mensajes, obtiene un número y el número de procesos en el grupo
  - 2. Ejecución: El proceso realiza operaciones interactuando con el resto por medio de las rutinas de paso de mensajes
  - 3. Finalización: Antes de terminar, el proceso debe desenrolarse del sistema de paso de mensajes



# Comunicación Punto a Punto



## Envío síncrono y asíncrono (condición de finalización)

### ■ **Síncrona:**

- El proceso que realiza el envío recibe información sobre la recepción del mensaje (fax)
- La comunicación se completa cuando el mensaje ha sido recibido

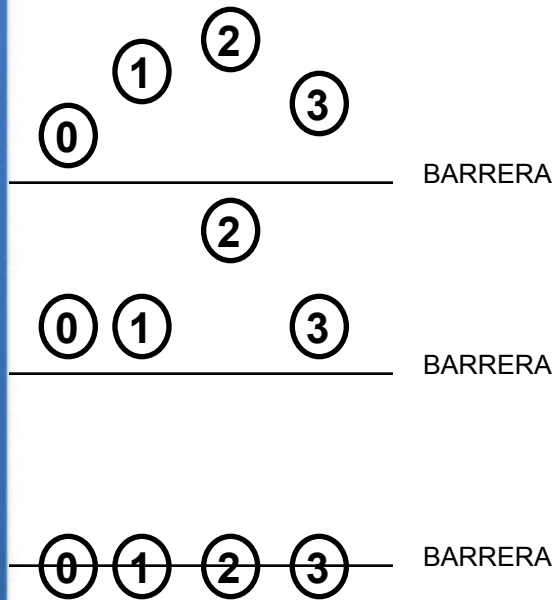
### ■ **Asíncrona:**

- El proceso únicamente conoce cuando se envía el mensaje (postal)
- La comunicación se completa tan pronto como el mensaje ha sido enviado

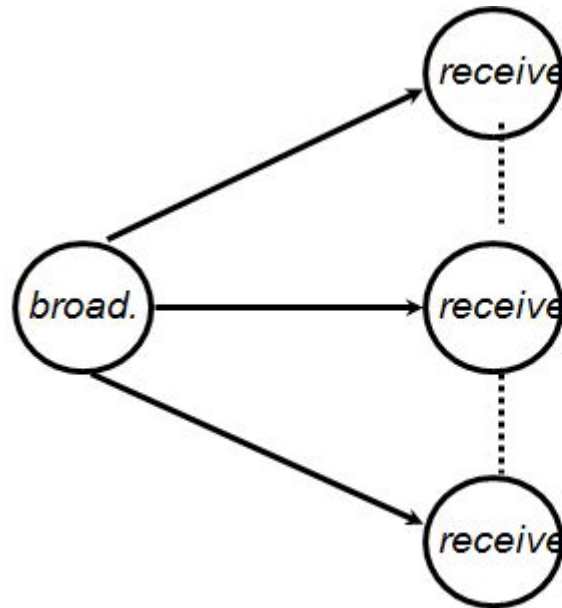
## Operaciones bloqueantes y no bloqueantes (espera o no la condición de finalización)

- **Operación no bloqueante:** Se inicia la operación y se vuelve al programa, por medio de otras funciones se puede comprobar la finalización de la operación
- **Operación bloqueante:** Solo se vuelve al programa cuando la operación ha finalizado

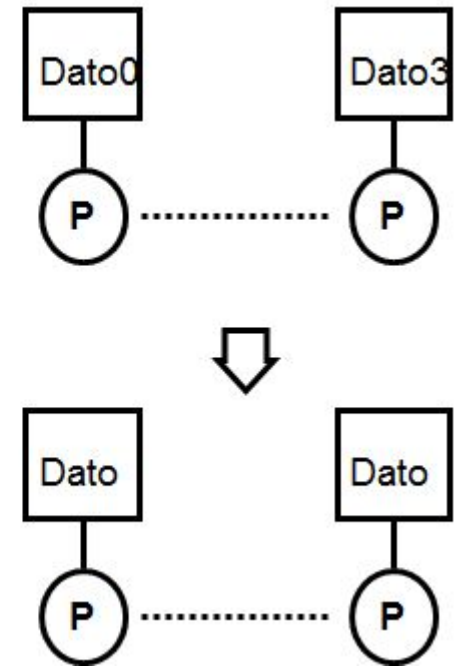
# Comunicaciones Colectivas



***Barrera***



***Broadcast***



***Reducción***

