

# T3: Principios de Programación Paralela


## T3.1: Conceptos

Departamento de Electrónica y Sistemas

Primavera 2016



# Índice

- 1 Introducción
  - 2 Niveles de Paralelismo
  - 3 Dependencias de Datos
- 

# Principios de Programación Paralela

## Objetivos del tema

- Adquirir principios básicos de programación paralela
- Conocer modelos de programación paralela
- Ser capaces de evaluar algoritmos paralelos
- Analizar posibilidades de paralelización
- Saber diseñar algoritmos paralelos
- Implementar códigos paralelos
- Técnicas de optimización de aplicaciones paralelas

# Introducción

## Computación secuencial por simplicidad

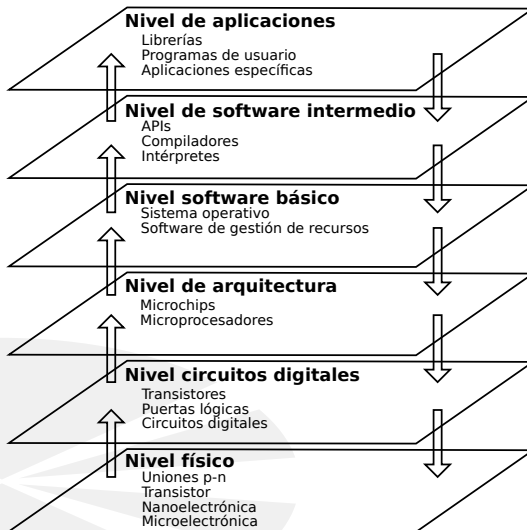
- **Modelo secuencial de computador** (Arquitectura von Neumann). Visión de alto nivel que considera el computador como un todo, en lugar de múltiples elementos operando simultáneamente.
- **Computación secuencial.** Ejecución de instrucciones en serie.
- Potencia computacional limitada.

## Limitaciones de la computación secuencial

- **Limitaciones físicas** (miniaturización y disipación de calor). Al límite de la ley de Moore: la densidad de transistores se duplicada cada 1,5 años desde 1965.
- Los 3 muros (consumo, memoria y paralelismo a nivel instrucción)
- Tamaño de problema (limitado por memoria direccionable)
- Requerimiento de tiempo real (limitado por potencia computacional)
- Grandes retos (e.g., análisis del genoma humano)

# Introducción

## Niveles en la estructura de un computador



# Introducción

## Potencia de la computación paralela

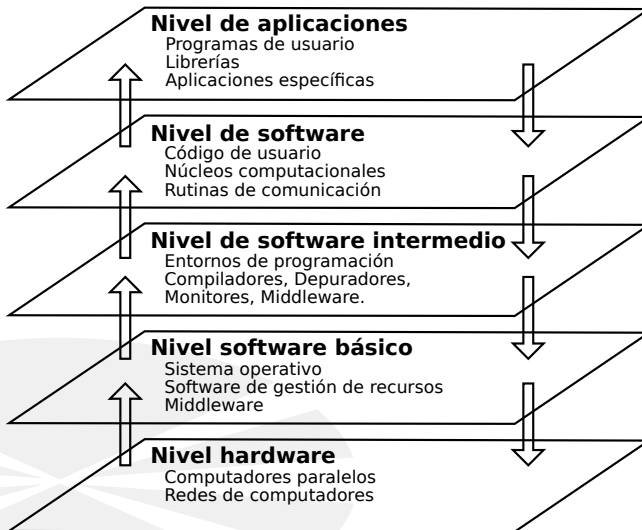
- **Computador paralelo.** Aquel que permite la ejecución simultánea de instrucciones.
- **Computación paralela.** Ejecución simultánea de tareas en un computador paralelo.
- Gran potencia computacional basada en la organización y no en la tecnología ya que es la misma.

## Aspectos de la computación paralela

- La **Computación de Altas Prestaciones** o High Performance Computing (HPC) se basa en la computación paralela.
- **Arquitecturas paralelas** desde un procesador dual-core hasta supercomputador TOP500 ([top500.org](http://top500.org)).
- **Popularización del paralelismo** debido a la ubicuidad de los procesadores multinúcleo.

# Niveles de Paralelismo

## Niveles en un computador paralelo



# Niveles de Paralelismo

## Niveles en un computador paralelo

- **Nivel hardware.** Replicación de recursos hardware (e.g., CPU cores, unidades funcionales, camino de datos, memoria), interconectados mediante buses/redes. Uso de aceleradores hardware (e.g., GPUs y FPGAs). Conceptualmente dos *modelos* (abstracciones) de computador paralelo:
  - *Computador paralelo de memoria compartida*
  - *Computador paralelo de memoria distribuida*
- **Nivel software básico.** Gestión del nivel hardware (SO, gestor de recursos, middleware)
- **Nivel software intermedio.** Herramientas para implementar aplicaciones paralelas (e.g., compiladores, librerías paralelas, depuradores, monitores, analizadores de rendimiento).



# Niveles de Paralelismo

## Niveles en un computador paralelo

- **Nivel software intermedio (cont.).** Dos aproximaciones:
  - *Compilador lenguaje paralelo o lenguaje secuencial con directivas paralelas* como OpenMP. (generalmente en computador paralelo de memoria compartida).
  - *Librerías de paso de mensajes* (generalmente computador paralelo de memoria distribuida).
- **Nivel de software.** Códigos o núcleos computacionales desarrollados por el usuario y ejecutables sobre un computador paralelo.
- **Nivel de aplicaciones.** Descansa sobre todos los demás niveles subyacentes. Así, una aplicación de simulación financiera puede depender de kernels numéricos computacionales, que a su vez pueden estar paralelizados con un entorno paralelo que incluye una librería de paso de mensajes, que se ejecuta a través de un SO/gestor de recursos sobre un computador paralelo.

# Dependencias de Datos

- **Dependencia de flujo (RAW o verdadera):** La instrucción I2 posee una dependencia de flujo respecto de I1, si una variable generada por I1 es utilizada como operando de I2. Dependencia “inherente” al código.
- **Antidependencia (WAR):** I2 es antidependiente de I1, si I2 modifica una variable utilizada como operando por I1.
- **Dependencia de salida (WAW):** I1 y I2 poseen una dependencia de salida entre ambas si las dos generan la misma variable.

## Detección de paralelismo entre dos segmentos de código

No tener dependencias es una condición que deben cumplir dos tareas para poder ser ejecutadas en paralelo.