

# Introducción a Open Multiprocessing (OpenMP)

# ¿Qué es OpenMP?

- **API** diseñada para la programación multiproceso en sistemas con arquitectura de memoria compartida, compatible con múltiples plataformas. Se compone de:
  - Directivas del compilador.
  - Biblioteca de funciones.
  - Variables de entorno.

**Nota:** No es un lenguaje de programación, sino que se basa en el modelo de hilos.

# OpenMP

Definido por un consorcio de proveedores de hardware y software, encabezado por AMD y conocido como OpenMP Architecture Review Board (ARB) desde 1997.

- BCS - Barcelona Supercomputing Center
- CAPS-Entreprise
- Convey Computer
- Cray
- Fujitsu
- HP
- IBM
- Intel
- Microsoft
- NEC
- NVIDIA
- Oracle Corporation
- Signalogic
- The Portland Group, Inc.
- Texas

Instruments Mas

información en

[www.openmp.org](http://www.openmp.org)

# ¿Qué es OpenMP?

- Modelo de programación paralela.
- Paralelismo basado en memoria compartida.
- Extensiones para lenguajes de programación existentes (C, C++, Fortran).
- Combina código serial y paralelo en un solo archivo fuente.

# Compiladores que soportan OpenMP

...

**GNU GCC** (a partir de la versión 4.2).

...

# Compilando con GNU GCC

El soporte para OpenMP ha de ser activado:

- Versión serial de un programa:
  - `gcc ejemplo.c -o ejemplo`
- Versión paralela de un programa:
  - `gcc -fopenmp ejemplo.c -o ejemplo`
- En la mayoría de los compiladores la opción es **openmp**

# Modelo de programación en OpenMP

- El paralelismo en OpenMP se especifica a través de directivas que se insertan en el código de C.
- Básicamente un programa en OpenMP es un programa secuencias que se añaden directivas OpenMP en el punto adecuado.
- En **lenguaje C** una directiva OpenMP tiene la forma:
  - **#pragma omp** <directiva OpenMP especificada>
- Todos los programas OpenMP deben incluir el archivo cabecera `#include <omp.h>`

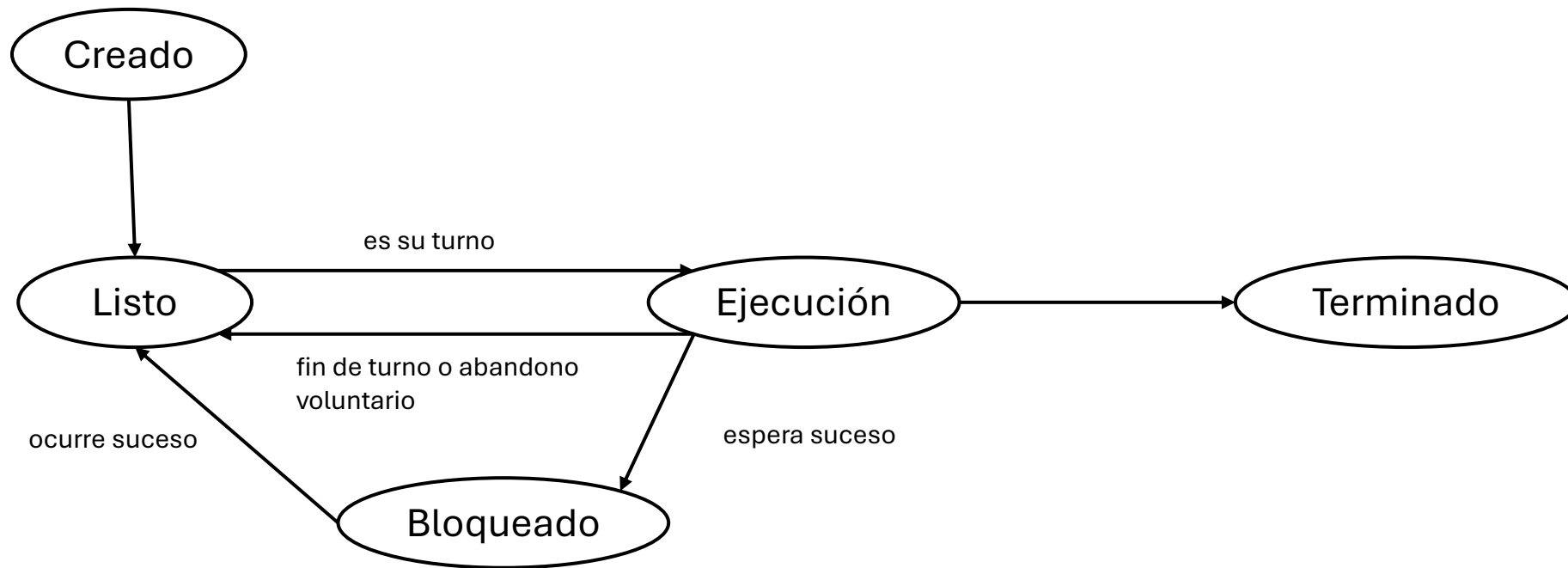
# Antes de iniciar....

1. ¿Qué es un proceso?
2. ¿Qué es un hilo?
3. ¿Qué se comparte entre los hilos?
4. ¿Cuál es la diferencia entre hilo y proceso?



# Proceso

Programa en ejecución que cuenta con su propio espacio de memoria y recursos, como CPU y memoria, para realizar tareas específicas.



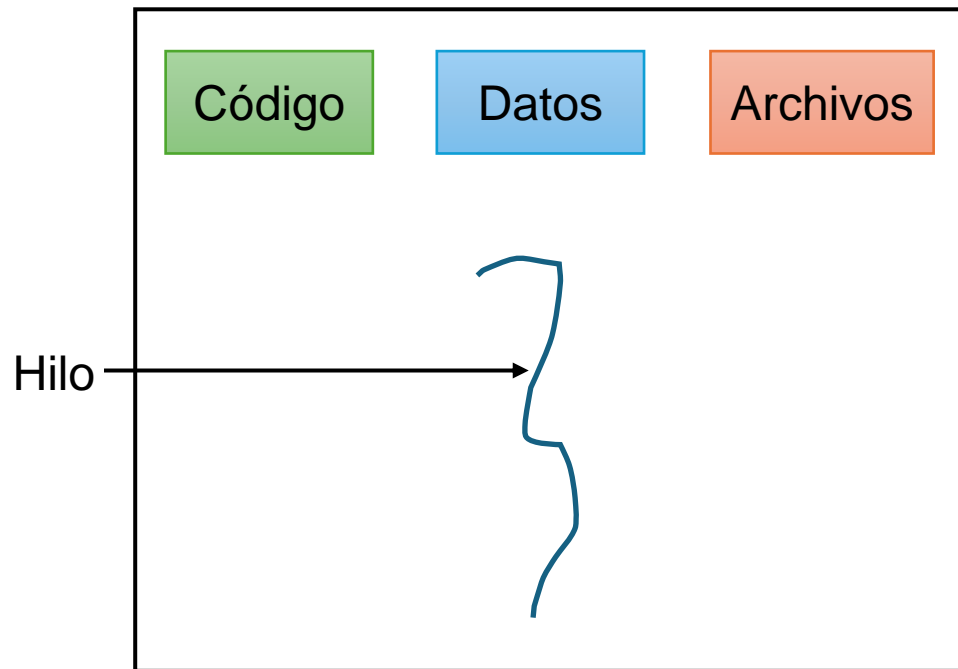
# Estados de un proceso

- En un principio, un proceso no existe.
- Una vez creado, el proceso pasa al estado denominado Listo (está en condiciones de usar la CPU tan pronto se le dé oportunidad).
- Un proceso puede pasar de Ejecución a Bloqueado cuando debe esperar a que ocurra un determinado evento o suceso, como:
  - La terminación de una operación de entrada/salida (E/S).
  - La finalización de otra tarea de otro proceso.

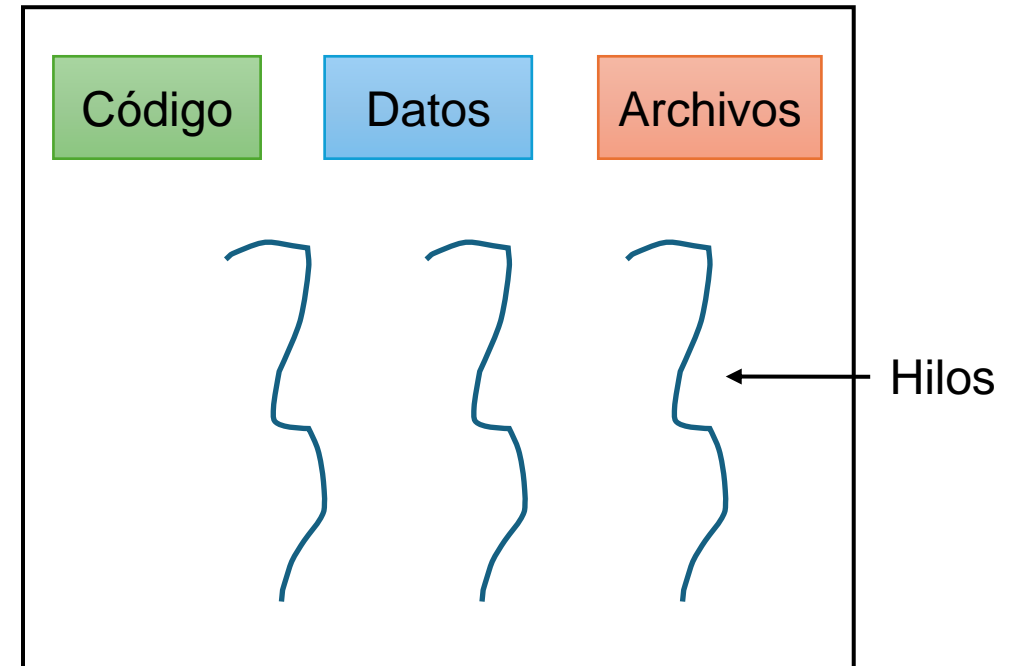
# Hilos (Thread)

- Dentro de un proceso puede haber varios hilos de ejecución.
- Un hilo, también conocido como hebra, proceso ligero, flujo, subprocesso o “thread”, es un programa en ejecución que comparte la imagen de la memoria y otros recursos del proceso con otros hilos.
- Por tanto, un hilo puede definirse como cada secuencia de control dentro de un proceso que ejecuta sus instrucciones de forma independiente.

# Procesos con un solo hilo y con múltiples hilos

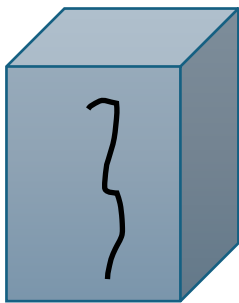


Mono-hilo

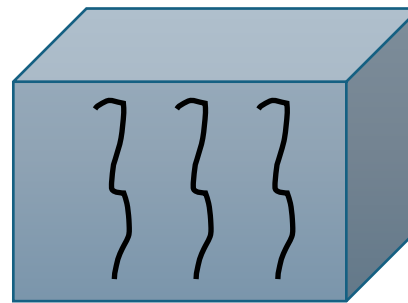


Multi-hilo

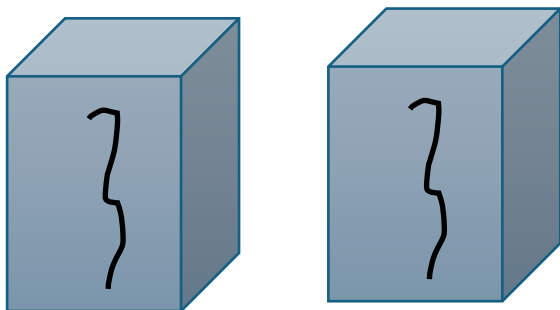
# Hilos y procesos



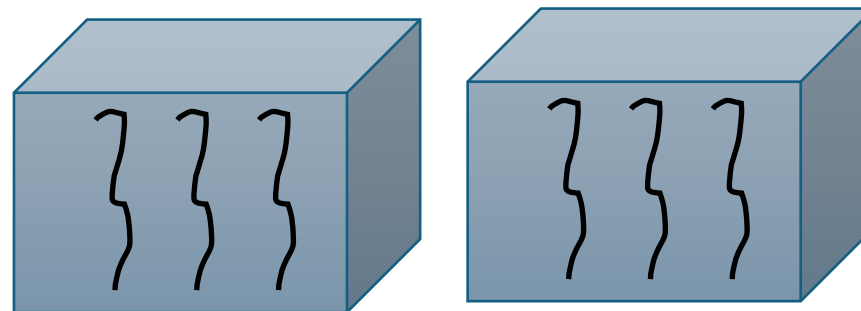
Un proceso, un hilo



Un proceso, varios hilos



Varios procesos, un  
hilo por proceso

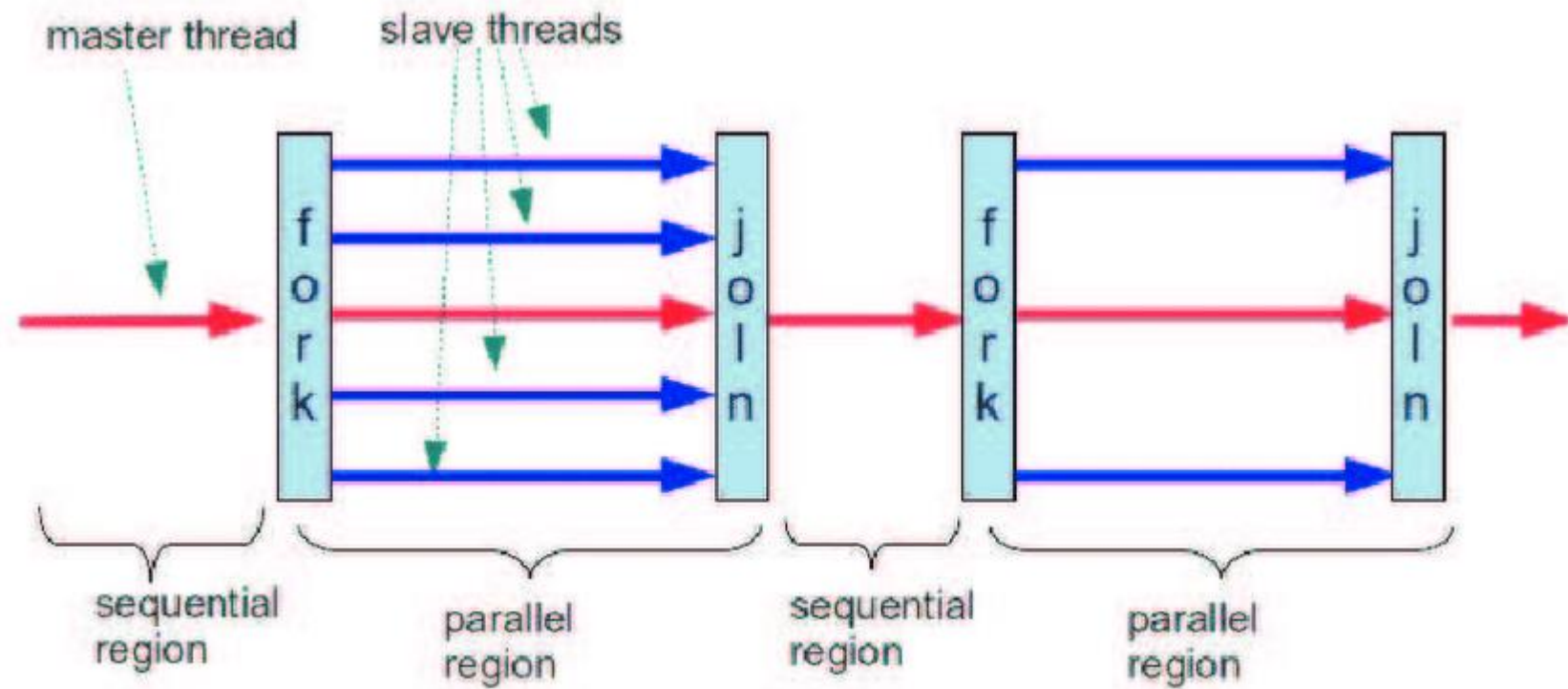


Varios procesos, varios hilos  
por proceso

# Arquitectura de OpenMP

- Fork / join (maestro - esclavo).
- Trabajo y datos compartidos entre hilos.
- Maneja sincronización (barreras, otras).

# Fork / Join



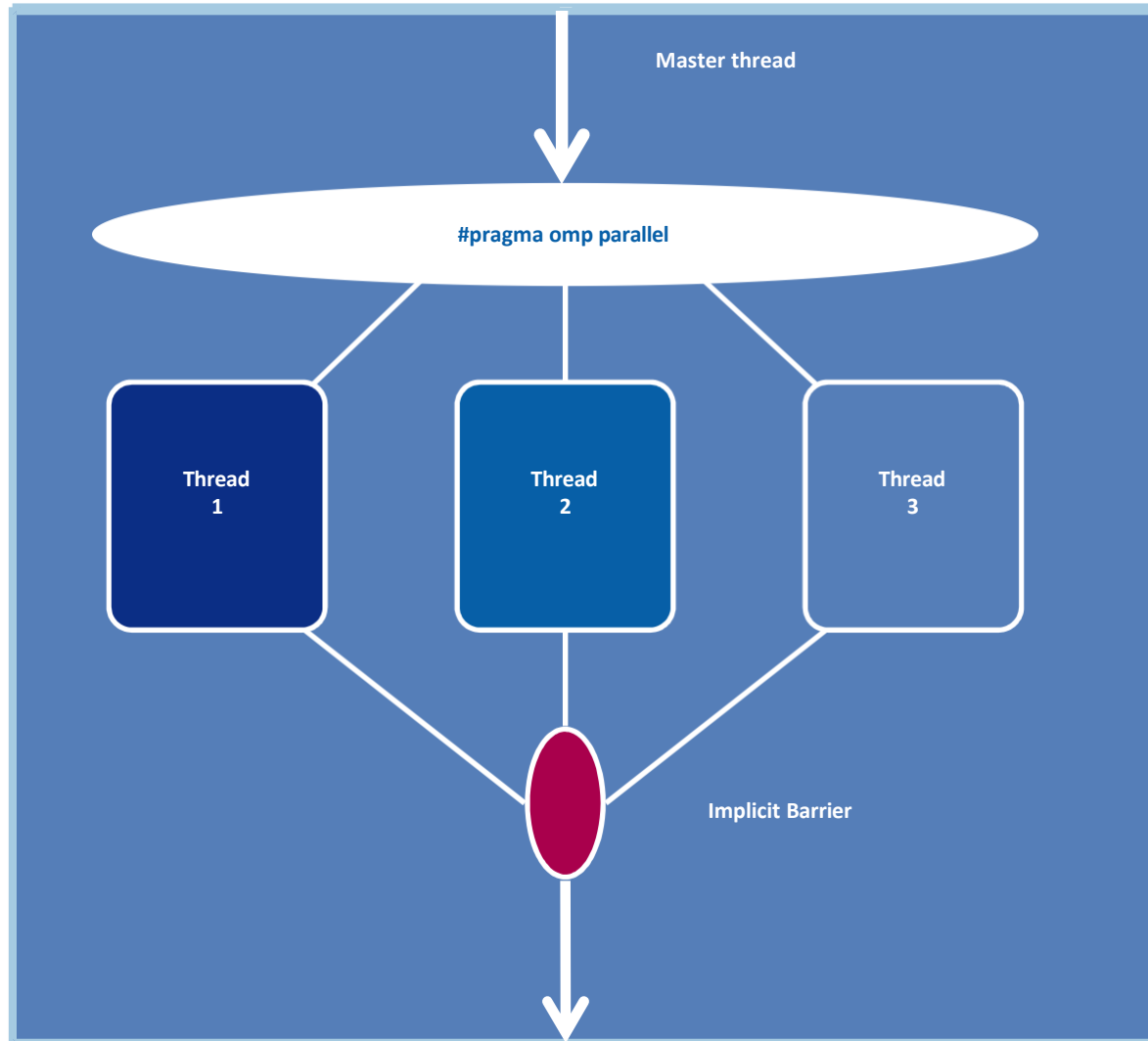
# Sintaxis

- Directivas o pragmas

- `#pragma omp construct [clause[clause]...]`
- Cláusulas: especifican atributos para compartir datos y calendarización.
- Una pragma en C o C++ es una directiva al compilador.



# Regiones paralelas



# Regiones paralelas

- Los hilos son creados desde el pragma *parallel*.
- Los datos son compartidos entre hilos.
- **C/C++:**

```
#pragma omp parallel
{
    bloque código
}
```

# ¿Cuántos hilos?

- Número de hilos = número de procesadores o núcleos.
- Intel lo usa de esta forma.
- Se definen más hilos con la variable de ambiente
  - OMP\_NUM\_THREADS.

# Actividad 1

- Compilar la siguiente versión serial:

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

```
int main (){
```

```
    int i;
```

```
    printf("Hola Mundo\n");
```

```
    for(i=0;i<6;i++)
```

```
        printf("Iteración: %d\n",i);
```

```
    printf("Bye");
```

```
}
```

# Actividad 1

- Agregar la directiva para ejecutar las primeras cuatro líneas del *main* en paralelo.
- Compilar con la opción `-fopenmp`
- ¿Qué sucede?

# Referencias

- Basado en: Sáenz García, E. K., & Valdez Casillas, O. R. (s. f.). *Programación paralela en OpenMP* [Diapositivas]. <http://lcomp89.fib.unam.mx/licad/assets/ProgramacionOpenMP/Programaci%C3%83%C2%B3nParalelaOpenMP.pdf>.