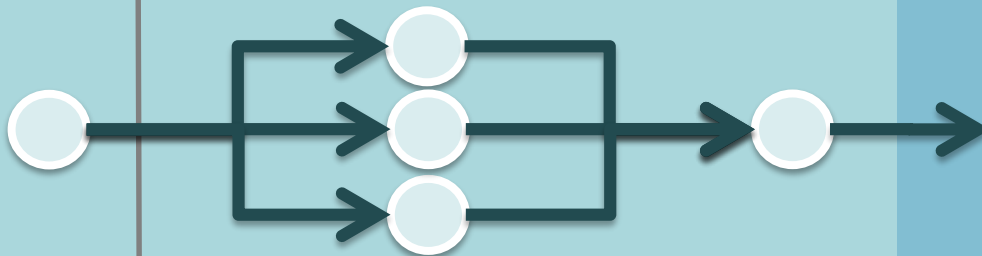


INTRODUCCIÓN A LA PROGRAMACIÓN PARALELA CON OPENMP (PARTE 1)



Computación de alto rendimiento

1. Introducción

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Índice

- ¿Qué es OpenMP?
- Diferencias entre concurrencia y paralelismo.
- Modelo de ejecución **Fork-Join** en OpenMP.

1. Introducción

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- ¿Qué es OpenMP?

- Es una **API** (Intefez de **P**rogramación de **A**plicaciones)
- **Open Multi-Processing**
- Es un Estándar **Abierto**, permite **P**rocesar **M**últiples hilos paralelos
- Se basa en directivas de compilador (`#pragma omp`)
- Funciona en **arquitecturas** de **memoria compartida**.
- Compatible con GCC, Clang, Visual Studio, Intel Compiler, entre otros.
- **Escalable**, permitiendo aumentar el número de hilos según el hardware disponible.

1. Introducción

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

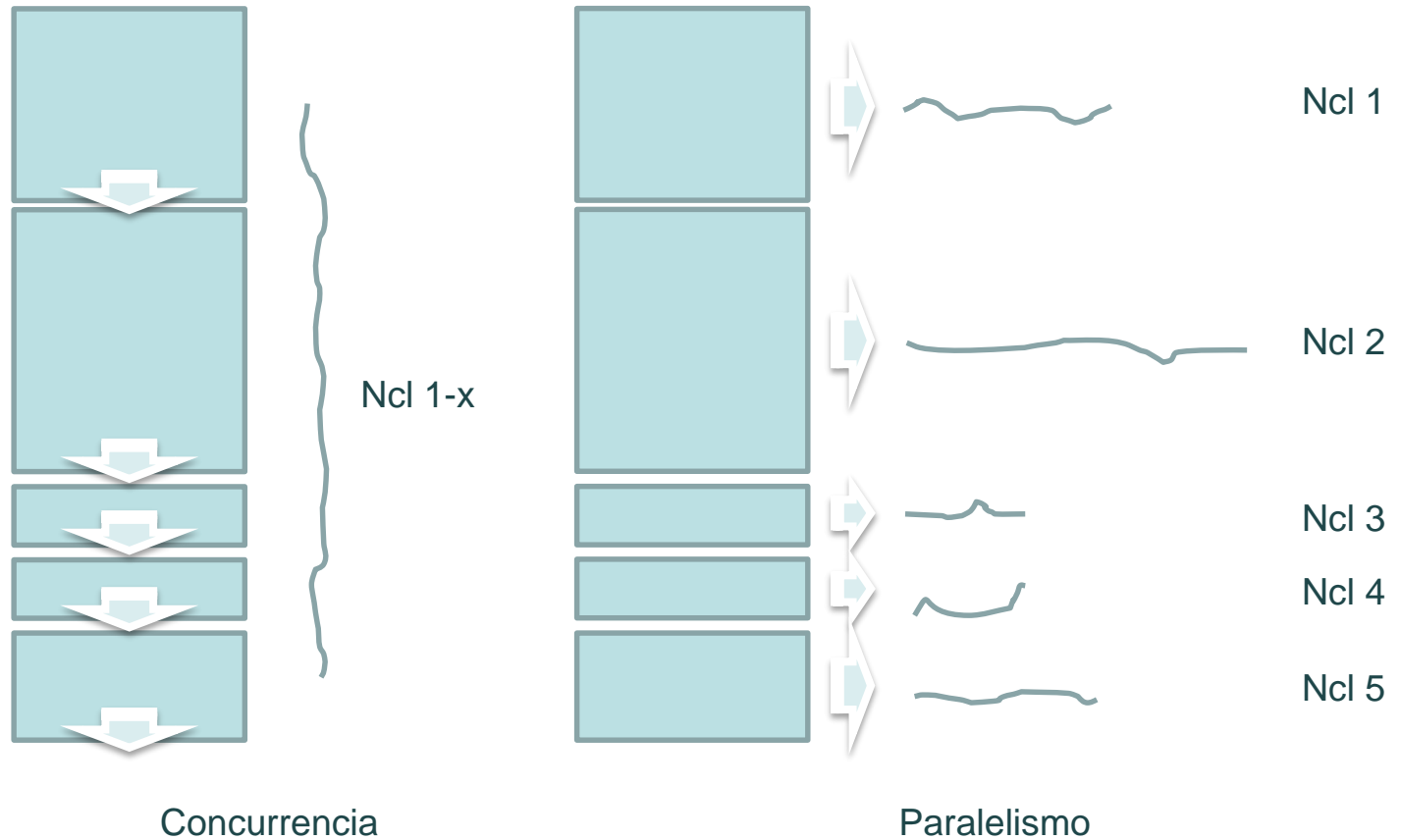
- ¿Qué es OpenMP?

- **Compatibilidades**

Plataformas compatibles	Lenguajes compatibles	Compilador
Linux, Windows, macOS	C, C++ y Fortran	GCC (GNU Compiler Collection)
Linux, Windows, macOS	C, C++	Clang/LLVM
Windows	C, C++	Microsoft Visual Studio (MSVC)
Linux, Windows, macOS	C, C++ y Fortran	Intel C++ Compiler (ICC, ICX)
Linux y sistemas IBM	C, C++ y Fortran	IBM XL Compiler
Linux, Windows	C, C++ y Fortran	PGI/NVIDIA HPC Compiler

1. Introducción

- Diferencias entre concurrencia y paralelismo.



1. Introducción

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Diferencias entre concurrencia y paralelismo.

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

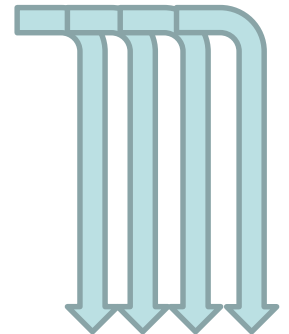
```
int main() {  
    for (int i = 0; i < 4; i++) {  
        printf("Hola desde la iteración %d\n", i);  
    }  
    return 0;  
}
```



Secuencial

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

```
int main() {  
    #pragma omp parallel for  
    for (int i = 0; i < 4; i++) {  
        printf("iteración %d del hilo %d\n", i, omp_get_thread_num());  
    }  
    return 0;  
}
```



Paralelo

Presentación de
prácticas

1. Introducción

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Diferencias entre concurrencia y paralelismo.

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

```
int main() {  
    for (int i = 0; i < 4; i++)  
        printf("Hola desde la iteración %d\n", i);  
}  
return 0;
```

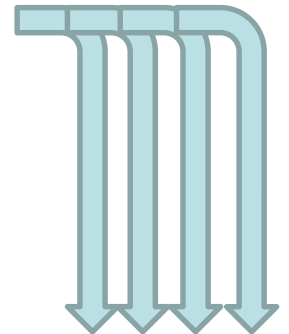
Hola desde la iteración 0
Hola desde la iteración 1
Hola desde la iteración 2
Hola desde la iteración 3



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

```
int main() {  
    #pragma omp parallel for  
    for (int i = 0; i < 4; i++) {  
        printf("iteración %d del hilo %d\n", i, omp_get_thread_num());  
    }  
    return 0;  
}
```

Hola desde la iteración 2 en el hilo 1
Hola desde la iteración 0 en el hilo 0
Hola desde la iteración 1 en el hilo 2
Hola desde la iteración 3 en el hilo 3

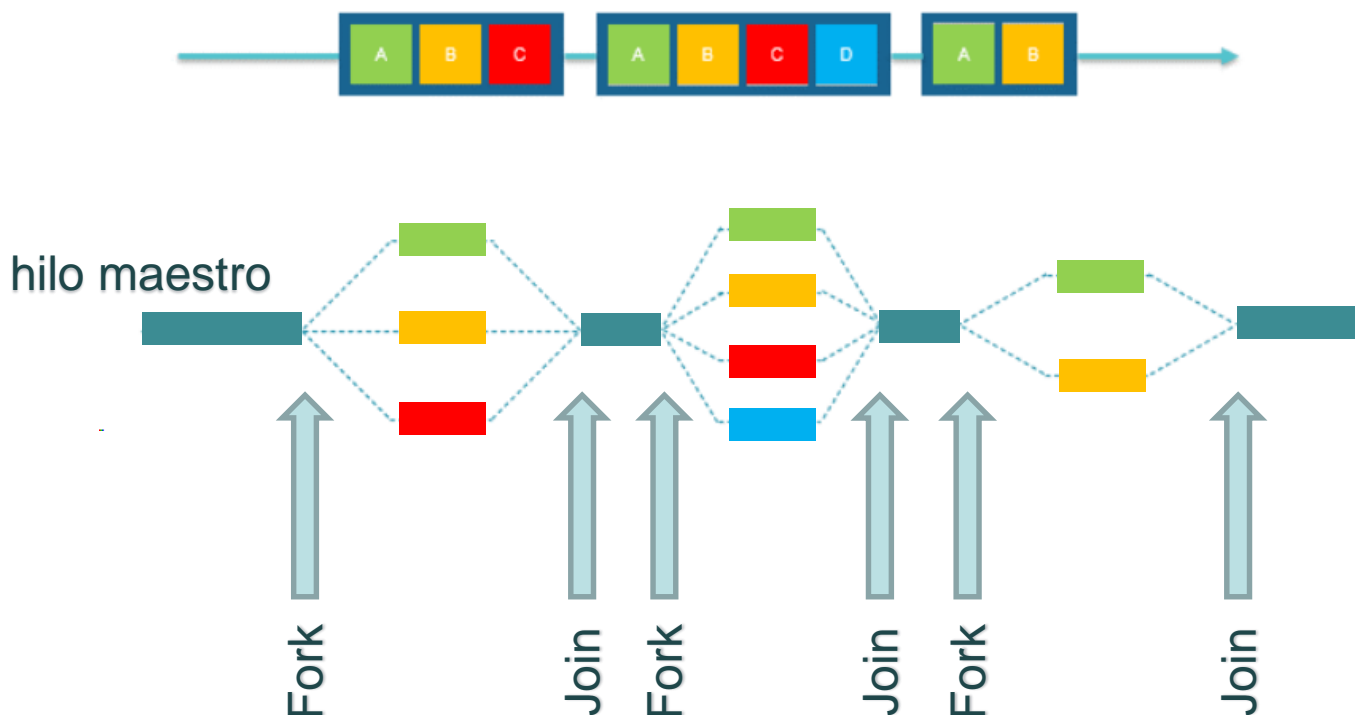


Presentación de
prácticas

⚠ **Nota:** El orden depende de cómo OpenMP asigne las iteraciones a los hilos en el momento de la ejecución.

1. Introducción

- Modelo de ejecución Fork-Join en OpenMP



Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Conclusión

Presentación de
prácticas

2. Configuración del Entorno

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Índice

- Instalación y uso de **OpenMP** en GCC, Clang y Visual Studio. (prácticas)
- Verificar si OpenMP está activo en tu computadora (prácticas)

2. Configuración del Entorno

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Instalación y uso de OpenMP en GCC, Clang y Visual Studio.

- Para usar OpenMP, es necesario activarlo en el compilador.
- Algunos ejemplos para diferentes compiladores y cómo activarlo:

Sistema Operativo	Compilador	Comando/Configuración
Linux/macOS/Windows (MinGW)	GCC	<code>gcc -fopenmp programa.c -o programa</code>
Linux/macOS/Windows	Clang	<code>clang -fopenmp programa.c -o programa</code>
Windows	Microsoft Visual Studio	Ir a Propiedades del Proyecto → C/C++ → Lenguaje . Activar Soporte de OpenMP (/openmp) .

Presentación de prácticas

2. Configuración del Entorno

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Verificar si OpenMP está activo en tu computadora
- Ejemplo.

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

int main() {
    #ifdef _OPENMP
        printf("OpenMP está disponible. Versión: %d\n", _OPENMP);
    #else
        printf("OpenMP NO está disponible en este compilador.\n");
    #endif
    return 0;
}
```

Presentación de
prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Índice

- `#pragma omp parallel` → Creación de hilos.
- `#pragma omp for` → Paralelización de bucles.
- `#pragma omp sections` → Dividir código en secciones ejecutadas por diferentes hilos.
- Variables compartidas (`shared`) vs privadas (`private`).
- Uso de `num_threads(n)` para definir el número de hilos.

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- `#pragma omp parallel` → Creación de hilos.
- Se usa para indicar que una región de código debe ejecutarse en paralelo.

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

int main() {
    #pragma omp parallel
    {
        printf("Este código se ejecuta en paralelo en el hilo %d\n", omp_get_thread_num());
    }
    return 0;
}
```

Presentación de
prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

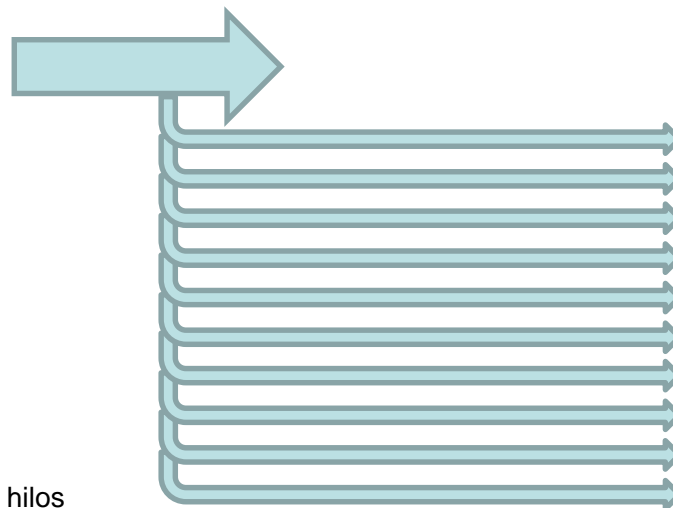
Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- `#pragma omp for` → Paralelización de bucles.
- Se usa para dividir automáticamente las iteraciones de un bucle entre los hilos (**ejemplo 1**)

```
#pragma omp parallel for schedule(static, 1) num_threads(10)
for (int i = 0; i < 10; i++) {
    printf("Iteración %d en el hilo %d\n", i, omp_get_thread_num());
}
```



Iteración 0 en el hilo 0
Iteración 1 en el hilo 1
Iteración 2 en el hilo 2
Iteración 3 en el hilo 3
Iteración 4 en el hilo 4
Iteración 5 en el hilo 5
Iteración 6 en el hilo 6
Iteración 7 en el hilo 7
Iteración 8 en el hilo 8
Iteración 9 en el hilo 9

Aquí:
10 iteraciones /10 hilos

Presentación de
prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

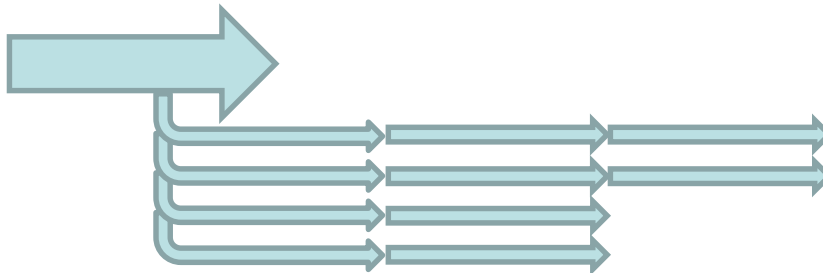
Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- `#pragma omp for` → Paralelización de bucles.
- **Ejemplo 2:**

```
#pragma omp parallel for
for (int i = 0; i < 10; i++) {
    printf("Iteración %d en el hilo %d\n", i, omp_get_thread_num());
}
```



Iteración 0 en el hilo 0
Iteración 1 en el hilo 1
Iteración 2 en el hilo 2
Iteración 3 en el hilo 3
Iteración 4 en el hilo 0
Iteración 5 en el hilo 1
Iteración 6 en el hilo 2
Iteración 7 en el hilo 3
Iteración 8 en el hilo 0
Iteración 9 en el hilo 1

Presentación de prácticas

Aquí:
10 iteraciones /4 hilos

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- `#pragma omp for` → Paralelización de bucles.
- **Ejemplo 3:** Supongamos que queremos sumar los números del 1 al 10 usando 4 hilos en paralelo.

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

int main() {
    int suma = 0;

    #pragma omp parallel for
    for (int i = 1; i <= 10; i++) {
        suma += i;
    }

    printf("Suma total: %d\n", suma);
    return 0;
}
```

- OpenMP divide el bucle **for** en partes y cada hilo se encarga de un grupo de iteraciones.
- **Problema:** Los hilos pueden modificar **suma** al mismo tiempo, lo que puede causar errores.

Presentación de prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Introducción a la clausula **Schedule()**
- Es una clausula que se usa con `#pragma omp for`
- Sirve para dividir tareas en bloques paralelos
- Hay tres tipos principales:
 - Static
 - Dynamic
 - Guided
- La elección correcta afecta al **rendimiento** y **balanceo de carga**
- Parámetro opcional: `chunk_size` determina el tamaño de los bloques

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

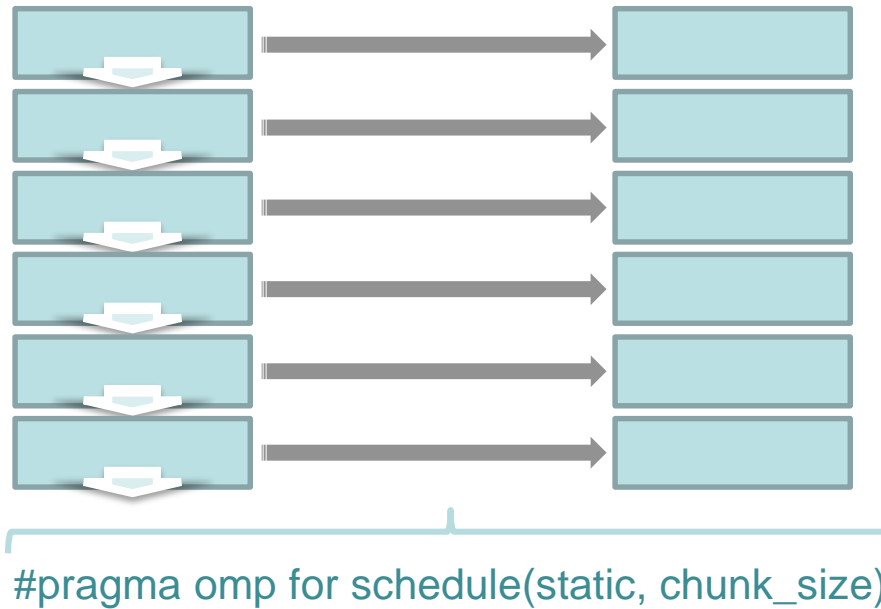
Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Schedule(**static**, chunk_size)
- Construye **bloques fijos** de n^0 aproximado **de iteraciones** a los hilos **antes de la ejecución**.
- Buena opción para **cargas homogéneas**
- **Menos sobrecarga** porque la distribución se hace de antemano



3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

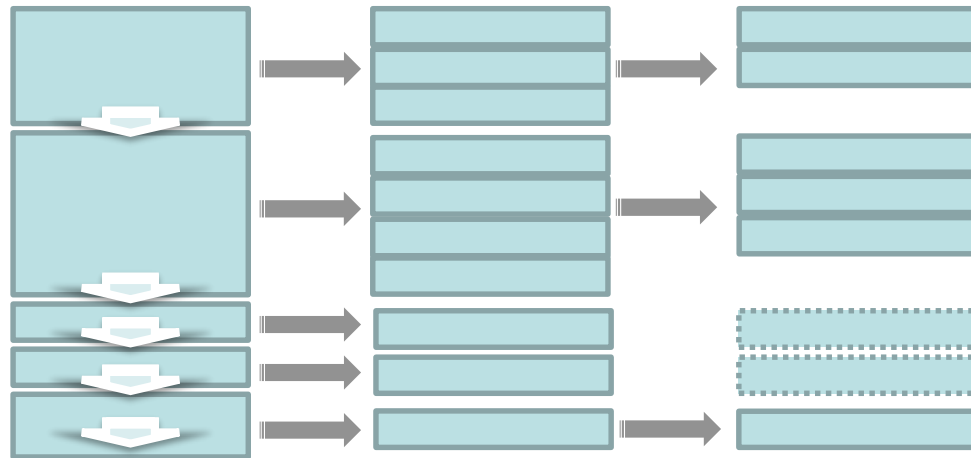
Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Schedule(**dynamic**, chunk_size)

- Asigna bloques de iteraciones **cuando un hilo queda libre**
- **Equilibra** la carga de trabajo cuando las iteraciones toman tiempos diferentes
- **Mayor sobrecarga** que `static` debido a la reasignación continua.
- **Ideal** cuando las iteraciones tienen **tiempos** de ejecución **variables**



```
#pragma omp for schedule(dynamic, chunk_size)
```

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

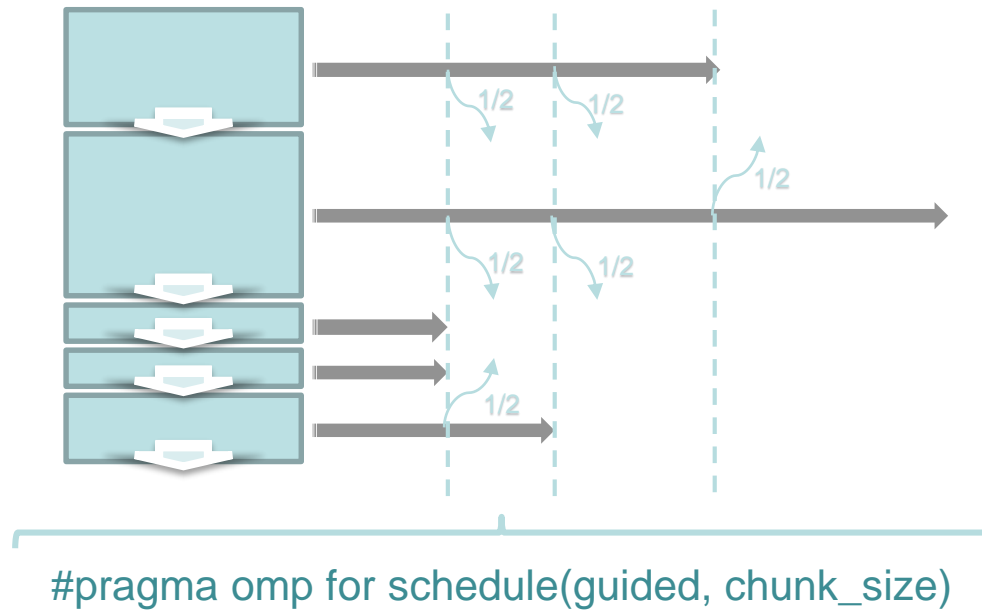
Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Schedule(**guided**, chunk_size)

- Similar a `dynamic`, pero **empieza con bloques grandes** y los reduce progresivamente
- Equilibra la carga y **reduce la sobrecarga administrativa** comparado con `dynamic`
- Buena opción cuando algunas **iteraciones son más pesadas** que otras



Presentación de
prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- ¿Cuándo usar cada una?
- **static** → Cuando todas las iteraciones toman **tiempo similar** y queremos la menor sobrecarga administrativa posible.
- **dynamic** → Cuando las iteraciones tienen **cargas muy desiguales** y queremos balancear la carga de trabajo.
- **guided** → Cuando las iteraciones tienen **cargas desiguales**, pero queremos **reducir el overhead** (tiempo administrativo extra) comparado con dynamic.

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- `#pragma omp sections` → Dividir código en secciones ejecutadas por diferentes hilos.
 - Si queremos que diferentes hilos hagan diferentes tareas, usamos `sections`.
 - La clasificación en tareas la decidimos manualmente por motivos del propio problema

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- `#pragma omp sections` → Dividir código en secciones ejecutadas por diferentes hilos.
- **Ejemplo:** En un motor de videojuegos, distintas tareas se ejecutan en paralelo:
 - 🎮 Renderizar gráficos
 - 🎵 Procesar sonido
 - 🎮 Detectar colisiones
 - 🧠 Calcular inteligencia artificial de los enemigos

Presentación de prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- `#pragma omp sections` → Dividir código en secciones ejecutadas por diferentes hilos.

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

```
void renderizar_graficos() { printf("🧠 Renderizando gráficos...\n"); }
void procesar_sonido() { printf("🔊 Procesando sonido...\n"); }
void detectar_colisiones() { printf("🎮 Detectando colisiones...\n"); }
void calcular_ia() { printf("🤖 Calculando inteligencia artificial de los enemigos...\n"); }
```

```
int main() {
    omp_set_num_threads(4);
    #pragma omp parallel sections {
        #pragma omp section renderizar_graficos();
        #pragma omp section procesar_sonido();
        #pragma omp section detectar_colisiones();
        #pragma omp section calcular_ia();
    }
    return 0;
}
```

Presentación de prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Variables compartidas (**shared**) vs privadas (**private**)
- **Shared** (compartidas):
 - La variable es la misma para todos los hilos.
 - Los hilos pueden leer y escribir en la misma dirección de memoria.
 - **Problema:** Puede causar condiciones de carrera si varios hilos escriben a la vez.
- **Private** (privadas):
 - Cada hilo tiene su propia copia de la variable.
 - Se inicializa sin valor definido al inicio del bloque paralelo.
 - Evita problemas de concurrencia, ya que cada hilo trabaja con su propia versión de la variable.

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Variables compartidas (shared) vs privadas (private)
- Ejemplo de uso:

```
int x = 10; // Variable compartida antes del bloque paralelo

#pragma omp parallel private(x)
{
    x = omp_get_thread_num(); // Cada hilo tiene su propia copia de x
    printf("Hilo %d, x = %d\n", omp_get_thread_num(), x);
}
```

Presentación de
prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Variables compartidas (**shared**) vs privadas (**private**)
- Ejemplo de problema sin **private** (Condición de carrera):

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

int main() {
    int x = 0;
    #pragma omp parallel num_threads(4) {
        x = omp_get_thread_num();
        printf("Hilo %d, x = %d\n", omp_get_thread_num(), x);
    }
    printf("Valor final de x: %d\n", x);
    return 0;
}
```

Dado que **x es compartida**, varios hilos la sobrescriben al mismo tiempo, causando comportamiento inesperado.

Presentación de prácticas

3.Directivas Fundamentales de OpenMP

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- Uso de `num_threads(n)` para definir el número de hilos.
 - **OpenMP** permite definir **cuántos hilos** queremos usar manualmente con `num_threads(n)`, en lugar de usar el valor por defecto.
 - Evito sobre cargar de hilos en sistemas con recursos limitados
 - Útil si la carga es previsible
 - Ejemplo:

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

int main() {
    #pragma omp parallel num_threads(4)
    {
        printf("Hilo %d de %d\n", omp_get_thread_num(), omp_get_num_threads());
    }
    return 0;
}
```

4. Ejemplos básicos

- Índice

- `#pragma omp parallel` → Creación de hilos.
- `#pragma omp for` → Paralelización de bucles.
- `#pragma omp sections` → Dividir código en secciones ejecutadas por diferentes hilos.
- Variables compartidas (`shared`) vs privadas (`private`).
- Uso de `num_threads(n)` para definir el número de hilos.

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

4. Ejemplos básicos

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- **Ejemplo básico de "Hola Mundo" con OpenMP**

- Código básico con `#pragma omp parallel`

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

int main() {
    #pragma omp parallel
    {
        printf("Hola Mundo desde el hilo %d de %d\n", omp_get_thread_num(), omp_get_num_threads());
    }
    return 0;
}
```

Salida esperada:

- La región paralela crea múltiples hilos, cada uno ejecutando el `printf()`.
- `omp_get_thread_num()` muestra el número del hilo.
- `omp_get_num_threads()` indica el total de hilos activos.
- El orden de la salida no está garantizado debido a la ejecución concurrente.

Presentación de
prácticas

4. Ejemplos básicos

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- **Paralelización de un bucle for simple con OpenMP**

- Código básico con `#pragma omp for`

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

int main() {
    #pragma omp parallel
    {
        #pragma omp for
        for (int i = 0; i < 10; i++) {
            printf("Iteración %d ejecutada por hilo %d\n", i, omp_get_thread_num());
        }
    }
    return 0;
}
```

Diferencias entre ejecución secuencial y paralela:

- **Ejecución secuencial:** Un solo hilo ejecuta todas las iteraciones del bucle.
- **Ejecución paralela:** OpenMP distribuye las iteraciones entre varios hilos.
- **Beneficio:** Ahorro de tiempo cuando el bucle tiene muchas iteraciones.

Presentación de
prácticas

4. Ejemplos básicos

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- **Medición del número de hilos con `omp_get_thread_num()`**

- Es útil saber qué hilo ejecuta cada parte del programa
- `omp_get_thread_num()`, devuelve el identificador del hilo actual

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

int main() {
    #pragma omp parallel
    {
        int id = omp_get_thread_num(); // Obtener el número del hilo
        int total_hilos = omp_get_num_threads(); // Obtener el número total de hilos
        printf("Hilo %d de %d hilos totales\n", id, total_hilos);
    }
    return 0;
}
```

```
Hilo 0 de 4 hilos totales
Hilo 1 de 4 hilos totales
Hilo 2 de 4 hilos totales
Hilo 3 de 4 hilos totales
```

Presentación de prácticas

El orden de la salida no está garantizado debido a la ejecución concurrente

4. Ejemplos básicos

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- **Comparación entre ejecución secuencial y paralela**

- Medición del tiempo de ejecución con `omp_get_wtime()`

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

int main() {
    double start, end;
    start = omp_get_wtime();

    #pragma omp parallel for
    for (int i = 0; i < 1000000; i++) {}

    end = omp_get_wtime();
    printf("Tiempo de ejecución: %f segundos\n", end - start);
    return 0;
}
```

Presentación de
prácticas

`omp_get_wtime()` mide el tiempo antes y después de la ejecución del bucle.

Se puede comparar el tiempo entre la versión secuencial y paralela.

5. Buenas Prácticas en Omp (parte 1)

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

Presentación de
prácticas

- **Cuándo y por qué paralelizar**
 - No siempre paralelizar mejora el rendimiento (por el overhead de creación de hilos).
 - Se recomienda en cálculos costosos o cuando hay suficiente trabajo para distribuir.
- **Overhead de la paralelización**
 - Creación y sincronización de hilos consume tiempo.
 - Evaluar si la carga de trabajo justifica el uso de OpenMP.
- **Ejemplos de errores comunes al paralelizar código**
 - Condiciones de carrera: Dos hilos modifican la misma variable sin sincronización.
 - Uso incorrecto de `private` y `shared`.
 - **Falso compartir:** Múltiples hilos acceden a datos en la misma línea de caché.
 - Uso incorrecto de `#pragma omp barrier` causando deadlocks.

6. Ejemplo Final

Introducción

Configuración

Directivas

Ejemplos Básicos

Buenas Prácticas

Ejemplos Finales

- Ejemplo completo

```
#include <omp.h>
#include <stdio.h>
#define N 1000

int main() {
    int i, suma_sec = 0, suma_par = 0, suma_dyn = 0, suma_guid = 0;
    double start, end;

    start = omp_get_wtime();
    for (i = 0; i < N; i++) suma_sec += i;
    end = omp_get_wtime();
    printf("Secuencial: %d, Tiempo: %f s\n", suma_sec, end - start);

    start = omp_get_wtime();
    #pragma omp parallel for reduction(+:suma_par) schedule(static, 10)
    for (i = 0; i < N; i++) suma_par += i;
    end = omp_get_wtime();
    printf("Paralela: %d, Tiempo: %f s\n", suma_par, end - start);

    start = omp_get_wtime();
    #pragma omp parallel for reduction(+:suma_dyn) schedule(dynamic, 10)
    for (i = 0; i < N; i++) suma_dyn += i;
    end = omp_get_wtime();
    printf("Dynamic: %d, Tiempo: %f s\n", suma_dyn, end - start);

    start = omp_get_wtime();
    #pragma omp parallel for reduction(+:suma_guid) schedule(guided, 10)
    for (i = 0; i < N; i++) suma_guid += i;
    end = omp_get_wtime();
    printf("Guided: %d, Tiempo: %f s\n", suma_guid, end - start);
    return 0;
}
```

Presentación de
prácticas