Programación Concurrente y Paralela (PCyP)

Dr. Mario Rossainz López
FCC- BUAP

- Introducción
- Conceptos y Definiciones
- Beneficios de la Programación Concurrente/Paralela
- Concurrencia y arquitecturas

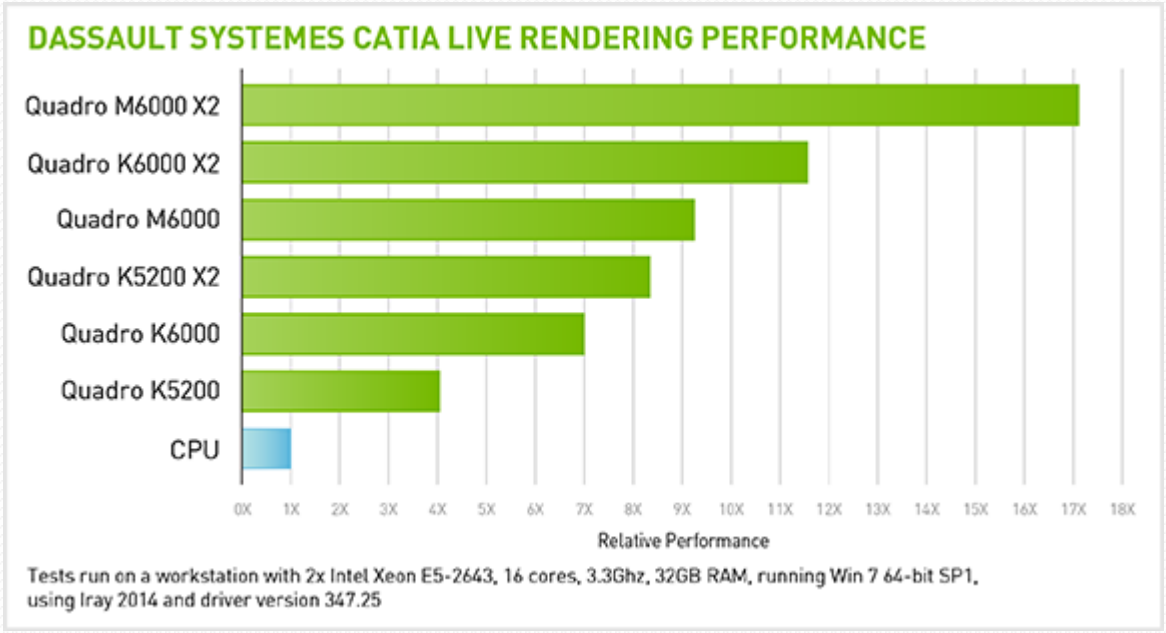
- Otoño 2022
- NRC: 60898

PROBLEMA

Solución de un problema
En una computadora con un
Sólo procesador (RENDIMIENTO)

Solución del mis problema
en una computadora con más de
un procesador (RENDIMIENTO)

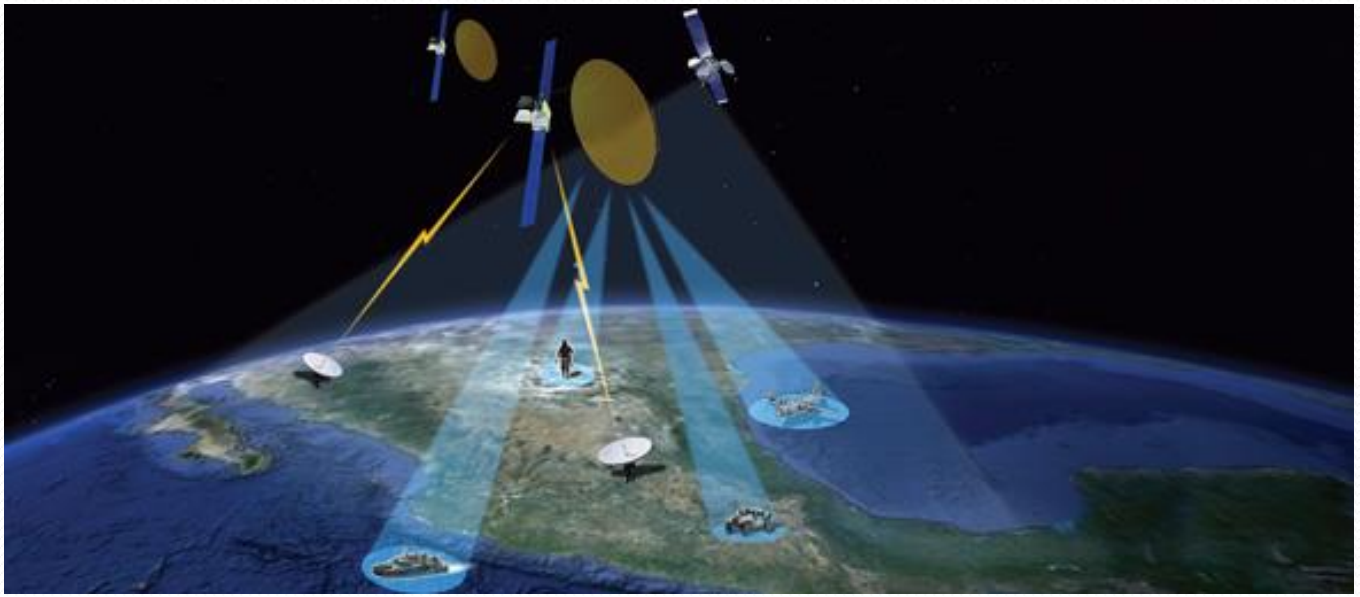
Mejora del RENDIMIENTO:
-Aplicaciones Paralelas
- Lenguajes de Programación Paralelos
- Arquitecturas Paralelas



Programación Concurrente y Paralela (PCyP)

Necesidad de la PCyP:

Ejemplo 1: Satélites recogiendo datos que informan sobre el clima, polución, agricultura y recursos naturales de la Tierra.



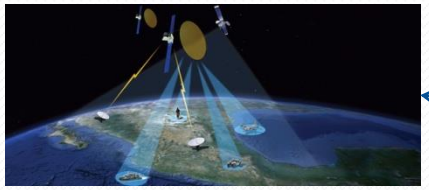
Programación Concurrente y Paralela (PCyP)

Necesidad de la PCyP:

Ejemplo 2: Grupo de médicos que requieren visualizar en una imagen tridimensional el cuerpo de un paciente que será intervenido y así ver secciones transversales de ciertos órganos, observar movimientos y simular la cirugía.

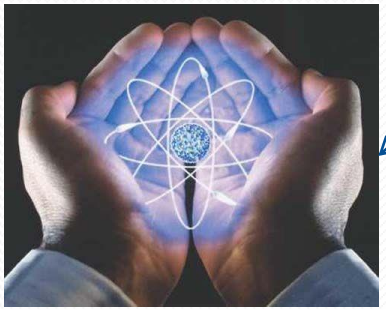
Programación Concurrente y Paralela (PCyP)

Satélites



Cirugías

Física Nuclear



Desarrollo de aplicaciones computacionales sobre computadoras rápidas para procesar enormes cantidades de datos.

Fármacos

Sismología

Aeronaútica

Programación Concurrente y Paralela (PCyP)

Conceptos y definiciones:

Procesamiento Paralelo: Enfoque computacional que ayuda a procesar datos y ejecutar un gran número de iteraciones de cálculo de una manera más eficiente dentro de las aplicaciones computacionales. Incluye el estudio de arquitecturas paralelas y algoritmos paralelos. Ej: computación heterogénea.

Paralelismo: Es la ejecución de un programa inicialmente secuencial en menos tiempo, utilizando para ello varios procesadores (más de uno).

Programa: Conjunto de instrucciones que dicen qué hacer con un conjunto de datos de entrada para producir algún tipo de salida.

Proceso: Programa en ejecución representado por el valor del contador del programa, el contenido de los registros del procesador, una pila y una sección de datos.

Programación Concurrente y Paralela (PCyP)

Introducción:

S.O.

Asociación

Prog. Concurrente

Repartición de tiempo
Entre muchos usuarios

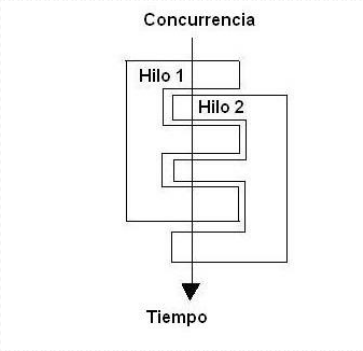
Sensación del usuario de que el procesador estaba dedicado para él

Programación Concurrente y Paralela (PCyP)

Introducción:

Hitos de la Programación Concurrente:

Aparición del concepto de Hilo o Thread

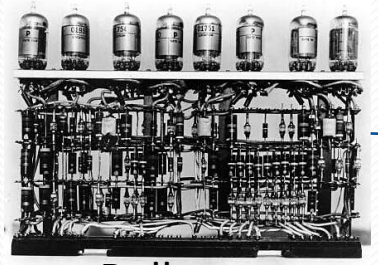


Aparición de lenguajes de programación con soporte de concurrencia

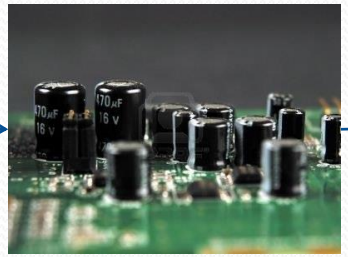
Aparición del Internet

Programación Concurrente y Paralela (PCyP)

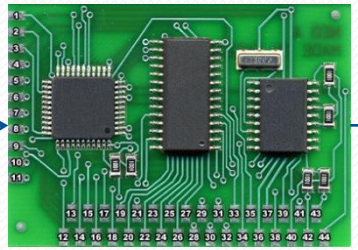
Necesidad de la PCyP:



Bulbos



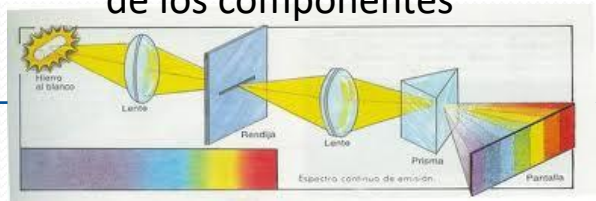
Transistores



Circuitos Integrados

VLSI

Factor de freno sobre el tamaño de los componentes



Velocidad de la luz en el vacío

Integración de Componentes

Ley de la física en cuanto a integración y acercamiento de dispositivos electrónicos que provoca disminución de velocidad y fiabilidad

Programación Concurrente y Paralela (PCyP)

Necesidad de la PCyP:

Posible solución: EL PARALELISMO (INCLUIDA LA CONCURRENCIA)

PARALELISMO

Se sabe que varias personas con habilidades similares terminan un trabajo en una fracción del tiempo que tardaría cada uno de ellos por separado.

Programación Concurrente y Paralela (PCyP)

Necesidad de la PCyP:

Posible solución: EL PARALELISMO (INCLUIDA LA CONCURRENCIA)

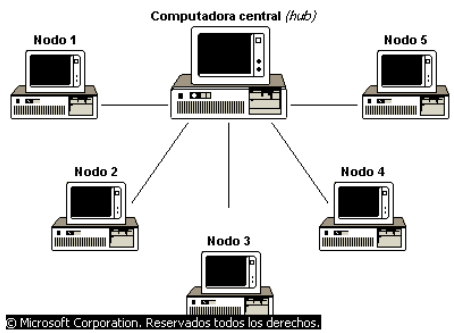
CONCURRENCIA

Se sabe que varias personas con habilidades similares terminan un trabajo en una fracción del tiempo que tardaría cada uno de ellos por separado.

Programación Concurrente y Paralela (PCyP)

Necesidad de la PCyP:

Posible solución: EL PARALELISMO (INCLUIDA LA CONCURRENCIA)



Programación Concurrente y Paralela (PCyP)

Necesidad de la PCyP:

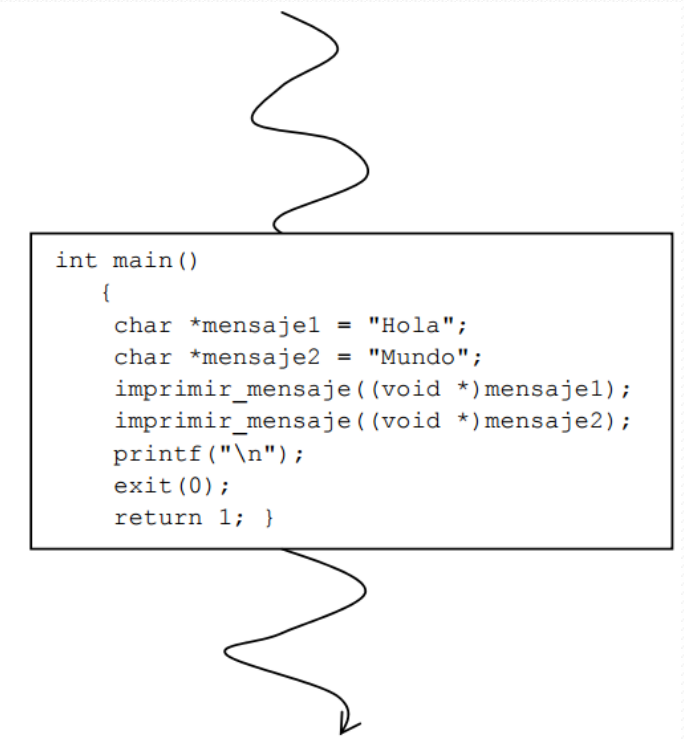
Posible solución: EL PARALELISMO (INCLUIDA LA CONCURRENCIA)

Finalmente, los componentes o hardware ya no es problema, ahora lo que hay que resolver es la forma de programar las máquinas de hoy. Se requieren entonces:

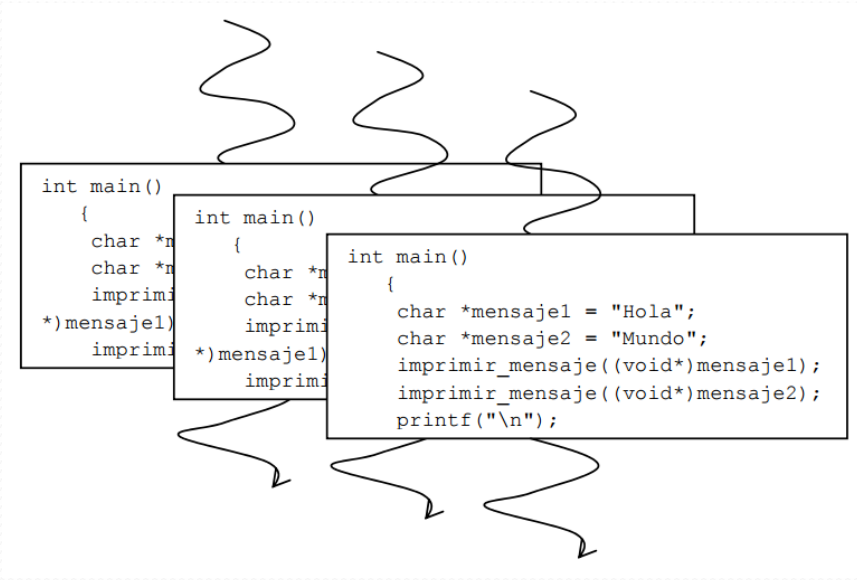
-Algoritmos
-Lenguajes
-Compiladores
-Sistemas operativos

Programación Concurrente y Paralela (PCyP)

Conceptos básicos de la PCyP:



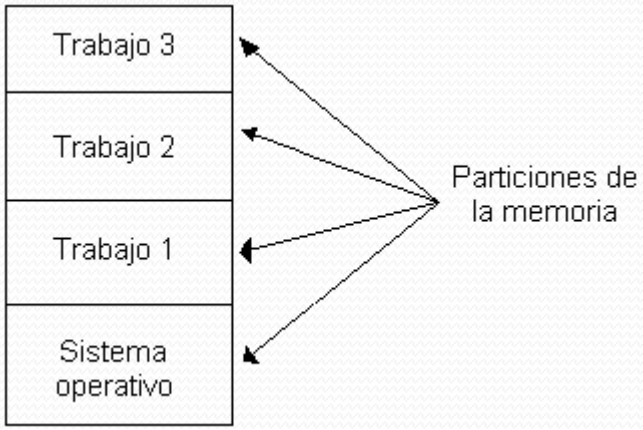
Un simple hilo de ejecución, Proceso ligero o Programa Secuencial



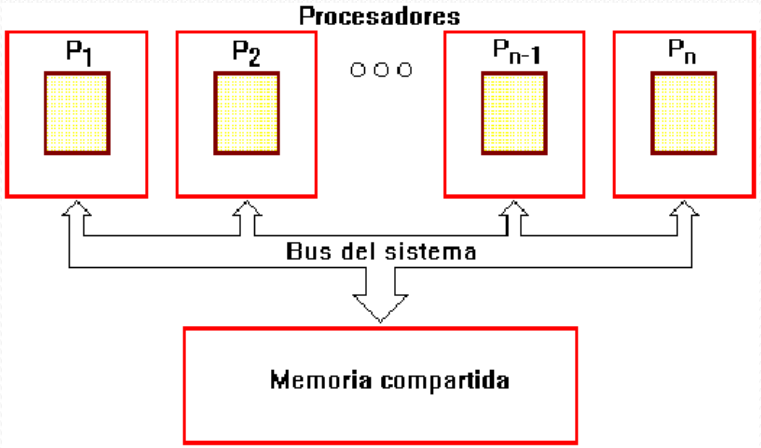
Múltiples hilos de ejecución trabajando en paralelo

Programación Concurrente y Paralela (PCyP)

Conexión física de los procesos:



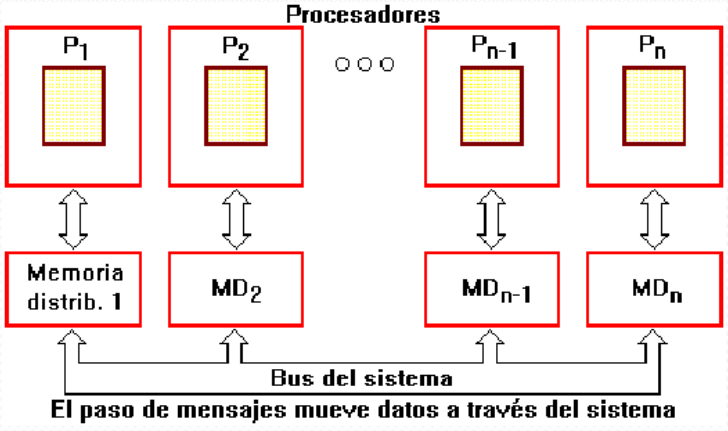
Multiprogramación: Ejecución Multiplexada en un solo procesador



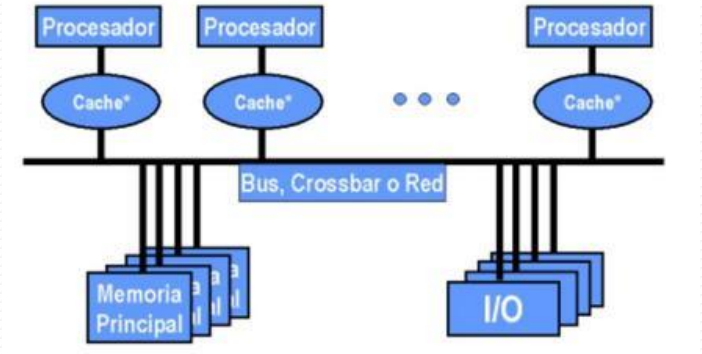
Multiproceso: Ejecución Multiplexada en un sistema multiprocesador de memoria compartida

Programación Concurrente y Paralela (PCyP)

Conexión física de los procesos:



Procesamiento Distribuido: Ejecución Multiplexada en varios procesadores Que no comparten memoria



Modelo Híbrido: Ejecución Multiplexada en varios procesadores Usando memoria compartida-distribuida

Programación Concurrente y Paralela (PCyP)

Programación Concurrente:

Definición: Conjunto de notaciones y técnicas utilizadas para describir mediante programas "el paralelismo potencial" de la solución de los problemas, así como para resolver los problemas de comunicación y sincronización que se presentan cuando varios procesos que se ejecutan de manera simultanea comparten recursos.

Propiedades:

1. NO DETERMINISMO: Dado un instante de tiempo, no es conocido que va a ocurrir en el instante siguiente (interrupciones al sistema, cambio de contexto, velocidad de ejecución no sincronizada de los procesadores, etc.).
2. Se requiere entonces de un modelo abstracto de concurrencia que permita saber cuando un programa es o no correcto (CORRECTITUD) independientemente de la ejecución del mismo.

Programación Concurrente y Paralela (PCyP)

Programación Concurrente:

Tipos de Interacción entre procesos:

1. Procesos Independientes: Aquellos que no se comunican entre sí y por lo tanto no requieren sincronizarse.

2. Procesos Cooperantes: Aquellos que colaboran en la realización de un trabajo común, y para ello deben comunicarse entre sí y sincronizar sus actividades.

3. Procesos en Competencia: Aquellos que comparten un número finito de recursos de un sistema computacional y deben "competir en una carrera" por obtener el uso de los recursos del sistema

Programación Concurrente y Paralela (PCyP)

Programación Concurrente:

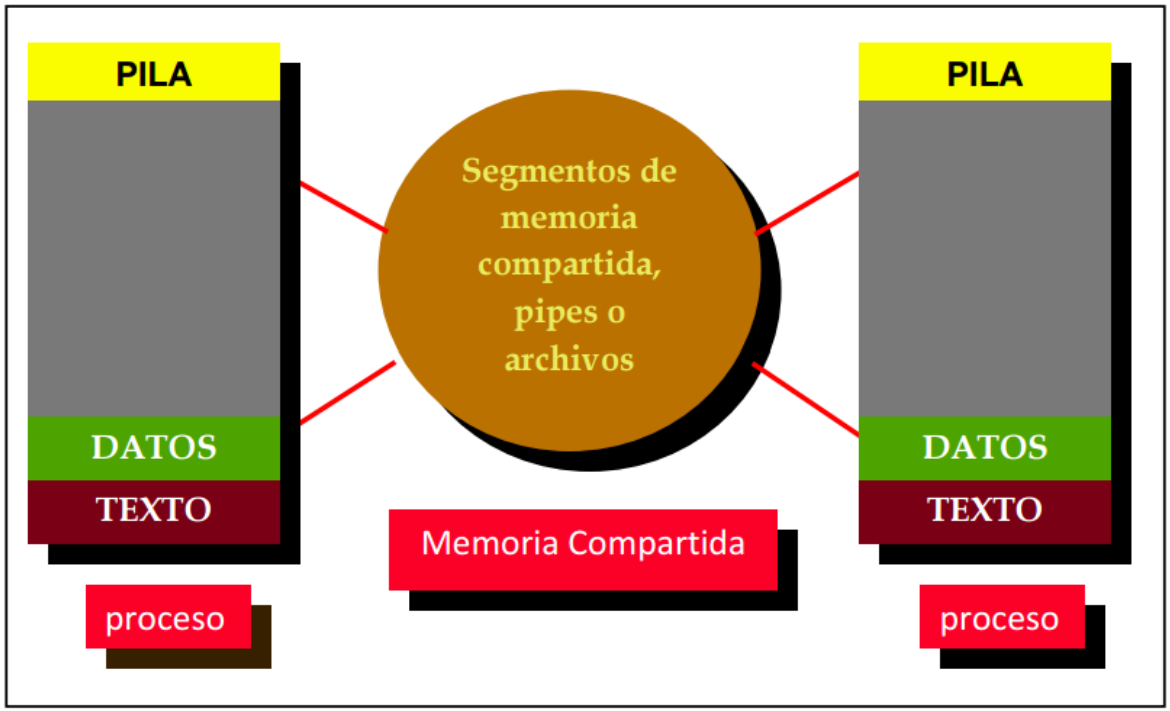
Propiedad de Abstracción:

1. Cada problema concurrente presenta un tipo distinto de paralelismo cuya implementación depende en principio de la arquitectura de la computadora en cuestión.
2. Para trabajar de forma independiente de la arquitectura se requiere utilizar un modelo abstracto de concurrencia que permita razonar sobre la correctitud de los programas que implementen el paralelismo con independencia de la máquina en donde se ejecuten.
3. Un lenguaje de programación es una abstracción que sirve al programador para comunicarse con la computadora sin tener que considerar la arquitectura física de la misma.
4. La Programación Concurrente como abstracción permite razonar sobre el comportamiento dinámico de los programas y sistemas paralelos cuando se ejecutan sobre un solo procesador [Paralelismo Abstracto]

Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

Proceso: Entidad compuesta de código ejecutable secuencial, un conjunto de datos y una pila.



Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

Variaciones de los Procesos:

1. ESTRUCTURA:
    1. **Estatica.-** Cuando el número de procesos del prog. Concurrente es fijo y se conoce en timpo de compilación.

    2. **Dinámica.-** Cuando los procesos pueden ser creados en cualquier momento. El número de procesos totales sólo se conoce en tiempo de ejecución.

2. NIVEL DE PARALELISMO:
    1. **Anidado.**- Cuando un proceso es definido dentro de otro proceso.

    2. **Plano**.- Cuando los procesos son definidos en el nivel más externo del programa.

Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

Variaciones de los Procesos:

3. RELACIONES ENTRE PROCESOS:
   1. **Padre/Hijo.-** Un proceso conocido como Proceso Padre es el responsable de la creación de otro llamado Proceso Hijo.

   2. **Guardián/Dependiente.-** Un proceso llamado Guardián no puede terminar su ejecución hasta que todos los procesos que dependen de él hayan terminado de ejecutarse.

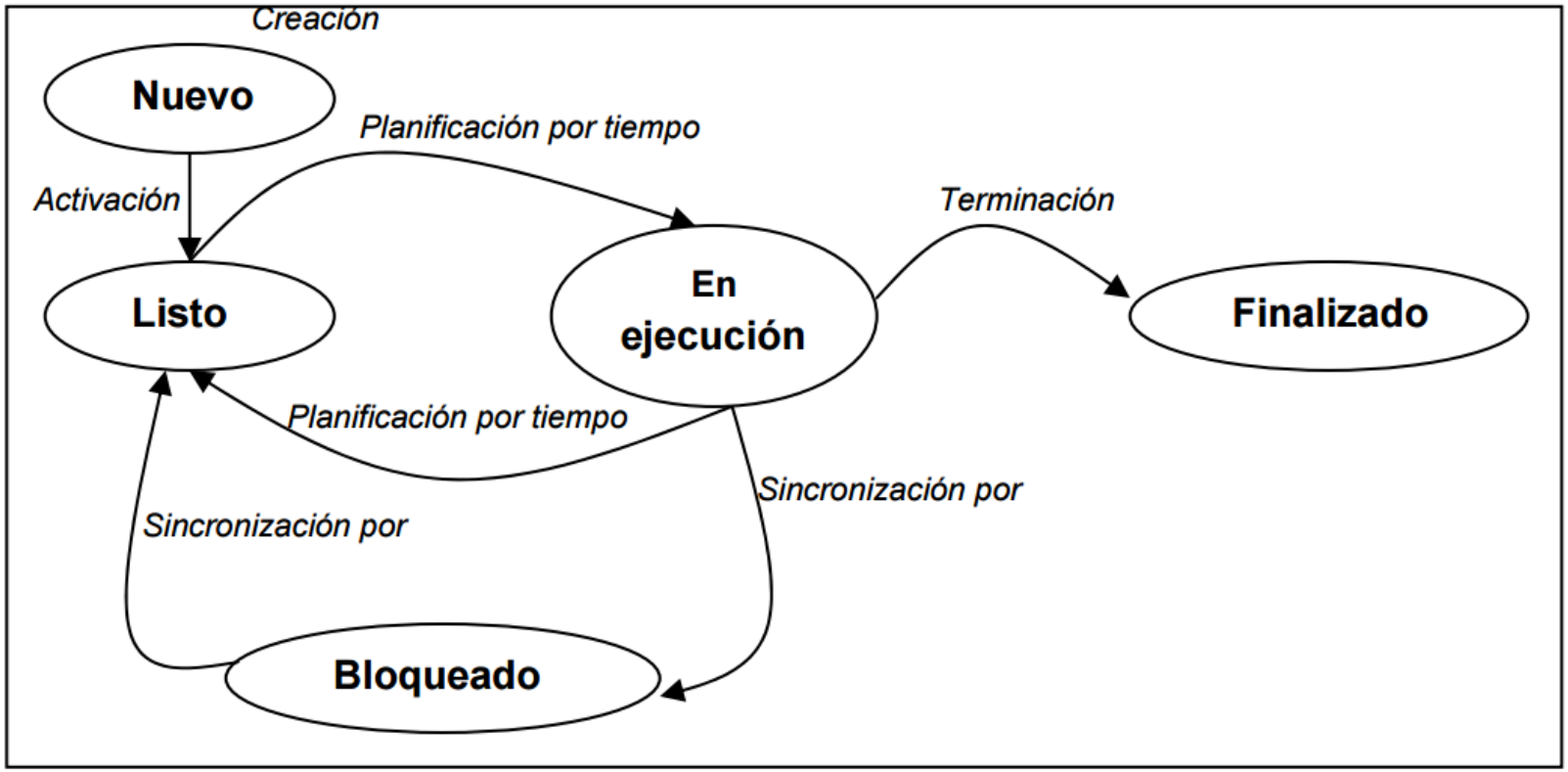
4. GRANULARIDAD:
   1. **Grano grueso.**- Cuando un prog. Concurrente tiene pocos procesos c/u con trabajo significativo por realizar.

   2. **Grano fino**.- Cuando un prog. Concurrente tiene un gran número de procesos, c/u de ellos con trabajo simple por realizar.

Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

Estados y Operaciones de un Proceso:



Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

Formas de crear y manejar procesos:

1. Llamadas al Sistema Operativo (Procesos Pesados)

2. Lenguajes de Programación que soportan Programación Concurrente, independientes del Sistema Operativo: Ada, Occam, Pascal Concurrente, Objective-C

3. Bibliotecas para creación y Operaciones con Procesos, independientes del Sistema Operativo: C, C++, C#, Java

Programación Concurrente y Paralela (PCyP)

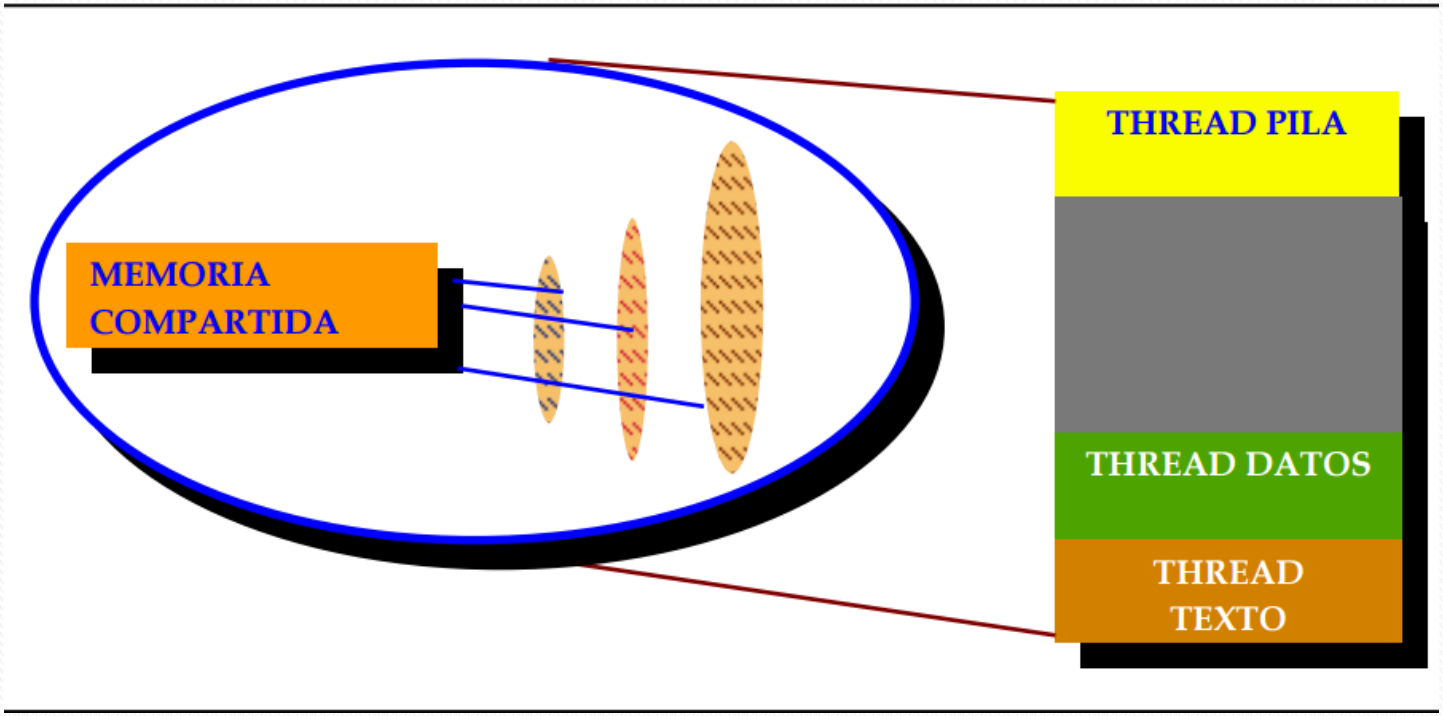
Procesos vs Hilos:

Hilos o Threads: Un hilo o Thread es una secuencia de instrucciones ejecutada dentro de un programa, es decir, cuando se ejecuta un programa el CPU utiliza el contador de programa del proceso o hilo para determinar qué instrucción debe ejecutarse a continuación.

MultiThreading: Es la ejecución de varios hilos dentro de un mismo proceso. A los hilos se les llama procesos ligeros y al proceso que los contiene se les llama proceso pesado.

Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:



Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

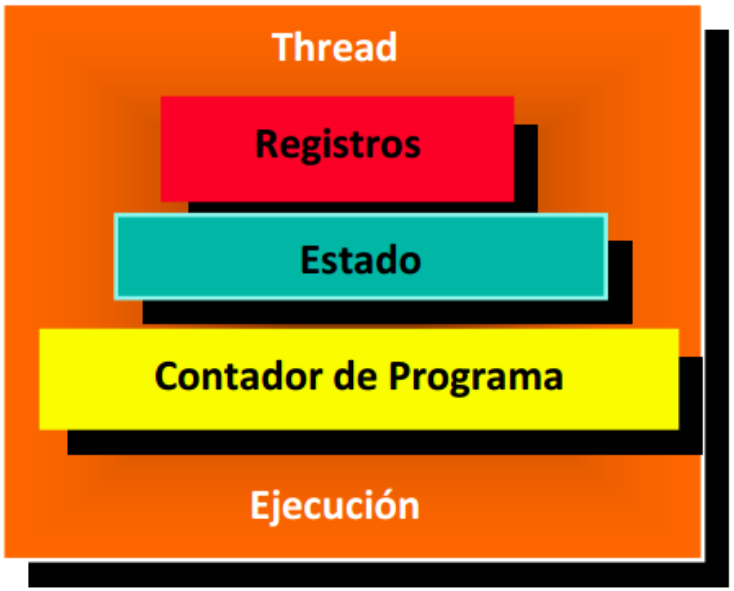


Figura 8. Modelo genérico de un Thread

Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

CARACTERÍSTICAS DE LOS HILOS:

1. Los hilos son más pequeños comparados con los procesos
2. La creación de un hilo es menos costosa que la de un proceso
3. Los hilos comparten los recursos mientras que los procesos requieren su propio conjunto de recursos
4. Los hilos son más económicos que los procesos al ocupar menos memoria
5. Los hilos proporcionan a los programadores la posibilidad de escribir aplicaciones concurrentes que se pueden ejecutar tanto en sistemas monoprocesador, como en sistemas multiprocesador de forma transparente.
6. Los hilos pueden incrementar el rendimiento de una aplicación en entornos monoprocesador

Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

PROGRAMACIÓN CON HILOS:

Todo lenguaje de programación que proporcione la posibilidad de programar de manera concurrente, debe ofrecer al programador instrucciones para:

- Crear
- Mantener
- Sincronizar
- Destruir

Hilos; ya sea mediante instrucciones propias o a través de la incorporación de una librería o biblioteca.

Programación Concurrente y Paralela (PCyP)

Procesos vs Hilos:

MODELOS MÁS POPULARES DE HILOS:

1. Match C Threads, CMU
2. Sun OS LWP Threads, SUN Microsystems
3. PARAS CORE Threads, C-DAC
4. Java-Threads SUN Microsystems
5. Chorus Threads, Paris
6. OS/2 Threads, IBM
7. Windows NT/95 Threads, Microsoft
8. POSIX-Thread, ISO/IEEE Estándar

POSIX-Thread es un estandar para hilos que se encuentra definido dentro del estandar formal internacional: POSIX 1003.1c-1995.

POSIX= Portable Operating Sistem Interface