

UNIVERSIDAD CATÓLICA DE PEREIRA

CONVERGENCIA EN SISTEMAS Y  
TELECOMUNICACIONES

## Apuntes de clase

*Juan David Garcia Acevedo*

Docente de la materia  
Rafael Ricardo RUBIANO

6 de agosto de 2024

# Índice general

<b>I</b>	<b>Procesamiento Paralelo</b>	<b>2</b>
<b>1.</b>	<b>Teoría</b>	<b>3</b>
1.1.	¿Qué es el Procesamiento Paralelo? . . . . .	3
1.1.1.	Definición . . . . .	3
1.2.	¿Qué es el Procesamiento Concurrente? . . . . .	4
1.2.1.	Definición . . . . .	4
1.3.	¿Qué es un Proceso? . . . . .	6
<b>II</b>	<b>Computación en la nube</b>	<b>8</b>
<b>III</b>	<b>Inteligencia Artificial</b>	<b>9</b>
<b>IV</b>	<b>IoT Intenet de las Cosas</b>	<b>10</b>
<b>V</b>	<b>Big Data y Data Analytics</b>	<b>11</b>
<b>VI</b>	<b>Metodología de Calificación</b>	<b>12</b>
<b>2.</b>	<b>Cortes</b>	<b>13</b>
2.1.	Porcentajes evaluativos . . . . .	13

## Parte I

# Procesamiento Paralelo

# Capítulo 1

## Teoría

### 1.1. ¿Qué es el Procesamiento Paralelo?

#### 1.1.1. Definición

##### Procesamiento Paralelo

El Procesamiento Paralelo es una técnica que divide grandes conjuntos de datos en piezas mas pequeñas, siendo esta técnica utilizada para acelerar la ejecución de tareas que requieren muchos cálculos.

Este procesamiento tiene algunas ventajas. En primer lugar tenemos el ejemplo anteriormente mencionado donde se puede acelerar la ejecución de tareas de muchos calculos, especialmente útil para trabajar con algoritmos de entrenamiento *Big data*. En segundo lugar, nos permite aprovechar Hardware y poder de máquina disponible, ya que permite el uso de distintos núcleos al mismo tiempo.

Así mismo, también hay desventajas y uno de sus mayores obstaculos es el desafío en la depuración, y la comunicación entre los diferentes núcleos ya que pueden causar un cuello de botella.

Uno de los mejores ejemplos para comprender el procesamiento paralelo, es cuando una familia va a realizar sus compras mensuales de alimentación a un supermercado, ¿Qué debe hacer la familia para llevarse sus productos al hogar?.

Deben hacer una **cola** para pagar por su comida, productos y demás acercandose a las cajas. Vemos que el supermercado cuenta con varias cajas para pagar, y **colas de espera** debido a la cantidad de personas que simultaneamente están realizando sus compras.

En la figura 1.1 podemos ver **Problem** como la comida y productos de la familia, por otro lado, tenemos las **Instructions** como las personas en la cola, vemos el **CPU** como multiples cajeras atendiendo a las distintas familias al mismo tiempo.

Vemos como una sola instrucción es ejecutada a la vez por un núcleo del **CPU**, aunque dispongamos en este caso de cuatro núcleos como nuestras cajeras.

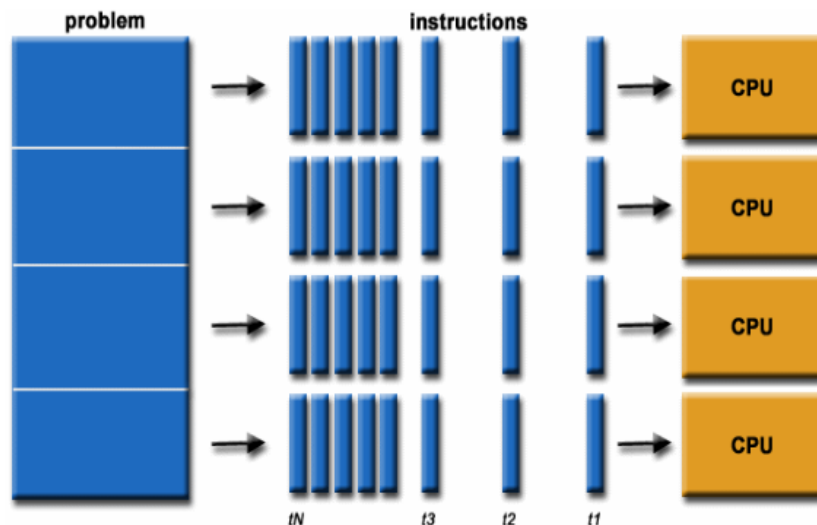


Figura 1.1: Ejemplo colas de supermercado, donde las **CPU** son las cajas donde cada familia, una por una procede a pagar por sus compras.

Cuando escribimos código, este en su proceso de compilación es ejecutado línea por línea.

Al igual que la figura 1.1 vemos el programa siendo ejecutado, es decir, se compila y de esta forma se convierte en un proceso el cual, línea a línea es enviada a un núcleo del procesador para ser ejecutado.

Podemos concluir que el **Procesamiento Paralelo** es un sistema con dos o mas procesadores conectados de tal manera que le sea posible compartir la ejecución de una determinada tarea.

## 1.2. ¿Qué es el Procesamiento Concurrente?

### 1.2.1. Definición

#### Programación Concurrente

Siguiendo con la idea del supermercado, en este ejemplo tenemos una tienda con una única caja, pero varias filas.

Partiendo de la figura 1.2 podemos ver **Problem** como la comida de la familia, y las **Instructions** como las personas en la cola. Y se ve el efecto de que un solo núcleo se ve asignado a un solo proceso.

Se asigna un **Quantum**(esto significa que se rotan por tiempo definido mediante un token, y este es asignado a cada uno de los procesos, para nuestro ejemplo las personas serían los procesos).

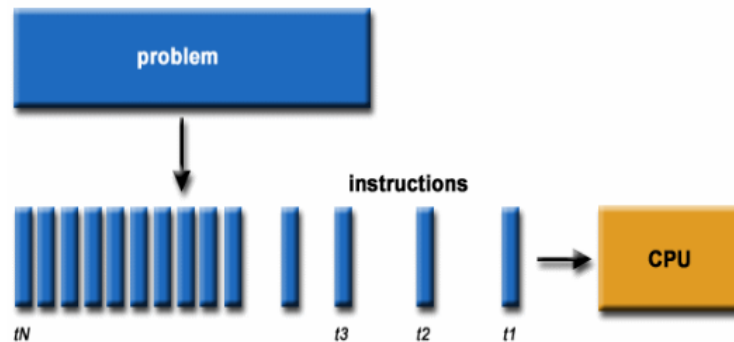


Figura 1.2: Ejemplo cola de supermercado, donde la **CPU** es la caja para pagar, y solo entra una familia a la vez a la caja.

Por cada línea de código se presenta una instrucción que se ejecuta a la vez. Cuando compilamos código, los buses conectan todas las partes de un computador al procesador, y se hace el proceso de conversión de Alto nivel a Bajo nivel, hablamos de la conversión de código como programa a proceso mediante su interpretación en binario. Esto es un proceso línea a línea.

Cuando hablamos de un computador tenemos:

- Partes
  1. Periféricos de entrada y salida I/O
  2. Central Process Unit
  3. Random Access Memory
  4. Buses de Datos

El CPU recibe las señales de los buses, siendo así la manera como el procesador recibe entradas y salidas temporales o permanentes.

Es importante aclarar sobre las arquitecturas de los procesadores, ya que existen de 32 bits y 64 bits.

Tenemos que las arquitecturas de 32 bits cuentan con 32 buses de datos para comunicarse con el **CPU**, y las arquitecturas de 64 bits, cuentan con 64 buses de datos.

Los procesadores como son máquinas hay que decirles cuando procesar, el **CPU** toma la información del disco que viaja mediante el bus en su velocidad de reloj para procesar, siendo esta la manera en que procesa la información del disco en el bus.

Para procesar, el procesador tiene unos transistores y trabajan con sistemas Flip-Flop guardando los estados de 1 y 0.

Estos son llamados Máquina de estado, donde se mueven los estados para conocer sus pulsos, y saber de esta manera entre los estados guardados la información que esta almacenada y escrita.

Cuando hay muchos procesos cada uno es interpretado instrucción por instrucción hasta que finalice, la solución se logra dando uso a varios procesadores. Por esto, el **Procesamiento Paralelo** es más rapido.

Luego tenemos el concepto *Data Center*, donde hay muchos nodos de computo, siendo cada uno de estos computadores con procesadores disponibles.

Un Data Center se comporta como una sola máquina, un solo servidor. Le podemos decir al Data Center que tenemos **10 Aplicaciones** durisimas, y esto supone un trabajo pesado en procesamiento, pero tenemos el Data center. Significa que podemos decirle a los procesadores que de manera **Paralela** se repartan la carga de la tarea. Por ejemplo, una carga de tarea pesada es el entrenamiento de un algoritmo de *Inteligencia Artificial*.

■ Conceptos importantes:

1. **Quantum**: Tiempo limite que es S.O asigna a a ejecución de un proceso, transcurrido este se pasa al siguiente.
2. **Cambio de contexto**: Proceso llevado a cabo por el S.O para salvaguardar los datos actuales de ejecución del proceso.

Para el sistema operativo ejecutar procesos a la vez, les asigna un **Quantum**, y en ese tiempo asignado se ejecutan las intrucciones, cuando se acabe debe llegar al siguiente proceso y ejecutar sus instrucciones. Cuando termine se vuelve a asignar un nuevo **Quantum**.

Para el sistema operativo salvaguardar los datos se utilizan punteros, cuando el Quantum finaliza se pone un puntero, "Se crea el Cambio de procesos se asigna un puntero para llevar el orden.

La programación concurrente no produce mayor velocidad, podemos decir que es más lenta, puesto que hay que esperar un turno para ejecutar las intrucciones, pero en paralelo cada una de las intrucciones tienen su núcleo. Permite la programación concurrente lidiar con varias tareas al tiempo.

El tiempo que se da al Quantum podría causar el efecto de lentitud y no se percibe el efecto de concurrente, y si es muy pequeño no se logra ejecutar las intrucciones.

De UNIX nacio la variante Linux, pero tenemos tambien Solaris y Windows, precisamente es difícil la creación de los sistemas operativos porque es difícil su depuración.

Si no esta el puntero de donde comienza, vuelve a comenzar desde cero, y estos punteros son guardados en la RAM.

### 1.3. ¿Qué es un Proceso?

Para un proceso poder ejecutarse requiere de espacio de direcciones. Uno de los procesos del sistemas operativos es el cambio de contexto. Los pro-

cesos tiene varios estados, y no siempre estan en ejecución, ya que pueden estar en espera de un subproceso.



## Parte II

# Computación en la nube

**Parte III**

**Inteligencia Artificial**

## Parte IV

# IoT Internet de las Cosas

## Parte V

# Big Data y Data Analytics

**Parte VI**

**Metodología de Calificación**

## Capítulo 2

# Cortes

### 2.1. Porcentajes evaluativos

#### **Cortes 1, 2, 3**

La metodología será para cada 15 días, ver las 2 horas por materia los días viernes.

- Corte 1

1. Talleres 10 %
2. Proyecto 10 %
3. Parcial 15 %
4. Acumulado 35 %

- Corte 2

1. Talleres 10 %
2. Proyecto 10 %
3. Parcial 15 %
4. Acumulado 35 %

- Corte 3

1. Talleres 10 %
2. Proyecto 10 %
3. Parcial 15 %
4. Acumulado 35 %