# Clase 01 - Introducción

## Definiciones

**proceso** = programa secuencial que conforma el programa concurrente

**multitasking** = coordinación de procesos secuenciales de forma paralela a lo largo del tiempo

**quantum** = tiempo que utiliza un proceso un procesador

**multitasking** = un procesador puede correr muchos programas (el scheduler se encarga de programarlos)

**multithreading** = un programa puede tener muchos hilos y ejecutarse cuando éste lo hace

**programa concurrente** = conjunto finito de procesos secuenciales

**proceso** = conjunto finito de instrucciones atómicas (se ejecutan en su totalidad o no se ejecutan)

## Colaboración entre procesos

para cumplir un mismo objetivo, mediante la

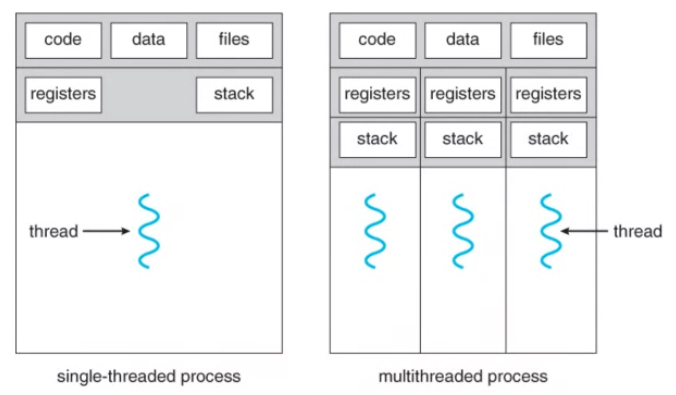
* sincronización → coordinar de forma temporal los procesos, operaciones tienen que cumplirse para que se cumplan otras; se sincronizan procesos secuenciales de un programa concurrente
* comunicación → los procesos se comunican datos para cumplir la función del programa concurrente

## Modelos de Concurrencia

* Estado Mutable Compartido → Se comparten regiones de memoria sincronizando los procesos que acceden a ella. La sincronización es propensa a errores.
* Paralelismo Fork-Join → Procesos independientes que se unen al final. Cada uno recibe una entrada y trabaja con ella, se une el resultado de cada uno a la salida. Los datos entre procesos se comunican mediante canales, no accesos a memoria (variables).
* Canales / Mensajes
* Programación asincrónica → concurrencia colaborativa, tareas que se ejecutan en un entorno que permita hacer más eficiente la ejecución, son tareas livianas y se busca que el programa luzca como un programa secuencial -utilizando concurrencia colaborativa-
* Actores → unicamente comunicación mediante mensajes o canales para su funcionamiento, intercambio de datos entre entidades (como POO)

## Threads

Unidades de ejecución, comparten el heap de un proceso. Más eficiente que compartir información entre múltiples procesos.



#### UNIX

Los trata como procesos a la hora de hacer scheduling.

# HW

* Principles of Concurrent & Distributed Programming (chaps 1 & 2)
* Operating System Concepts (chap 4)