

# PROGRAMAS MULTIPLES

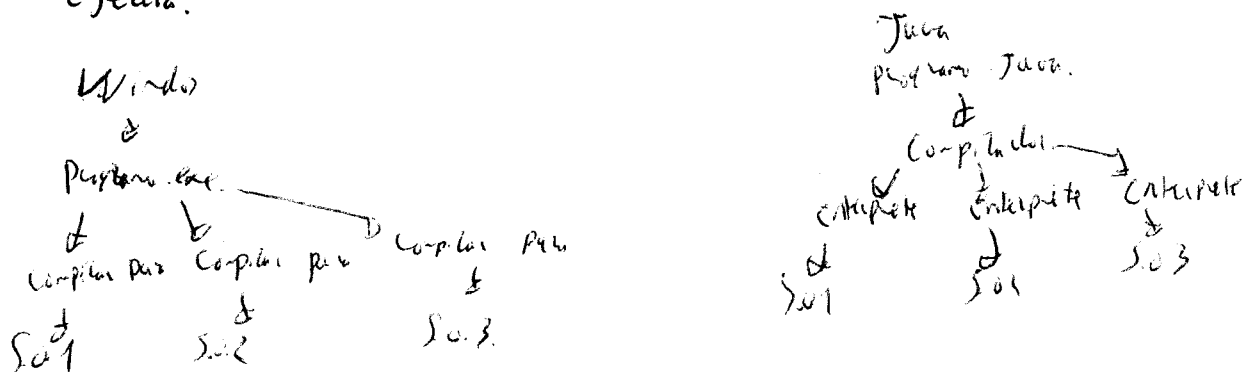
¿qué es un ejecutable?

Un ejecutable es un archivo que contiene instrucciones que puede interpretar el Sistema Operativo.

En Windows son los archivos .exe y están pensados para funcionar en el S.O. y pueden funcionar en cualquier proceso / pero no tiene que estar optimizado.

Java es un archivo .jar. Java que no son archivos ejecutables.

El intérprete de Java (Java.exe) lee el código y lo ~~ejecuta~~ ejecuta.



En Java solo hay que compilar un programa una vez, después para cada intérprete de Java que se produce para mi S.O. y además optimiza lo optimiza para mi configuración.

¿Qué es un proceso?

¿qué es un proceso?

~~Un proceso~~ es un ejecutable que se encuentra en ejecución dentro del proceso. Estado de un proceso.

• En ejecución. Estado del proceso.

• En pausa. A la espera de continuar.

• Interrumpido. Se termina el proceso antes de acabar ~~pero puede~~ puede interrumpirse la ejecución.

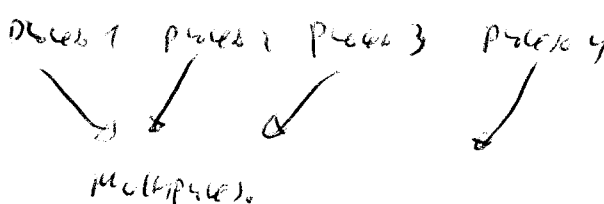
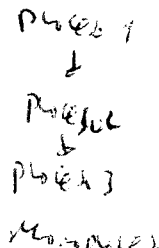
## ¿qué es un servidor?

- es proceso que se ejecuta en línea plan (no interacción en el tiempo).  
De forma (servicio) a otros procesos.

## ¿qué es un hilo?

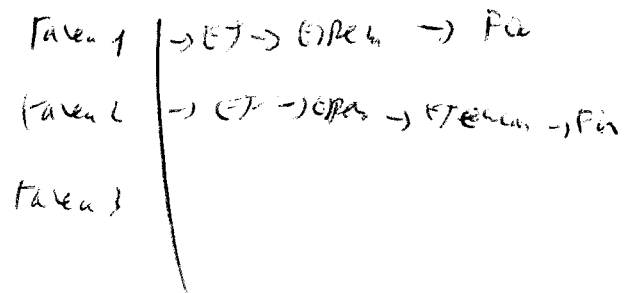
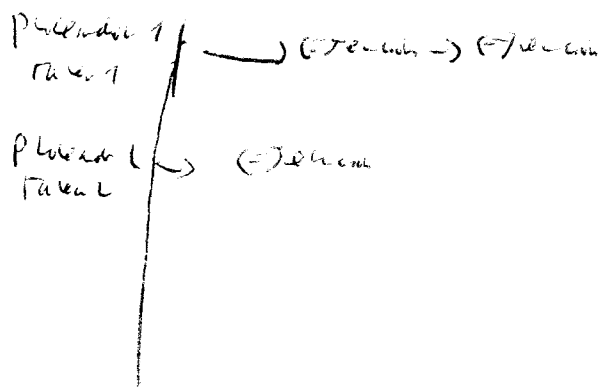
- En un tipo de procesos muchos hilos que comparten los datos de forma distinta a los procesos.
- un proceso no tiene acceso a los datos de otro proceso un hilo tiene acceso a los datos de su hilo.

## SISTEMAS MONOPROCESADOR Y MULTIPROCESADOR

- Sistema monoprocesador: Solo ejecuta un proceso. Como un solo.
- ~~Sistema multiprocesador~~
- Sistema multiprocesador: Puede ejecutarse varios procesos de forma simultánea.  


## SISTEMAS MULTIPROCESADOR

- Ejecución de forma simultánea en varios procesos. Se puede hacer por un hilo varios procesadores o por un procesador ~~por~~ hilos de forma multiprocesador.



# PROCESADOR

Componente del hardware del ordenador.

La CPU (Central Processing Unit).

Un procesador puede tener varios núcleos, que dan distintos niveles de hardware.

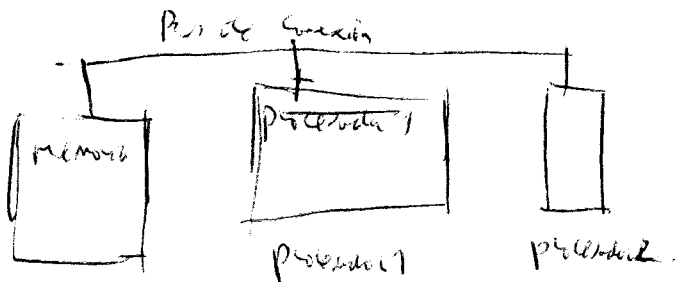
Cada procesador (núcleo) se puede dividir en nivel binario de base para procesadores reales (por ejemplo).

## ACOPLO DE LOS PROCESADORES

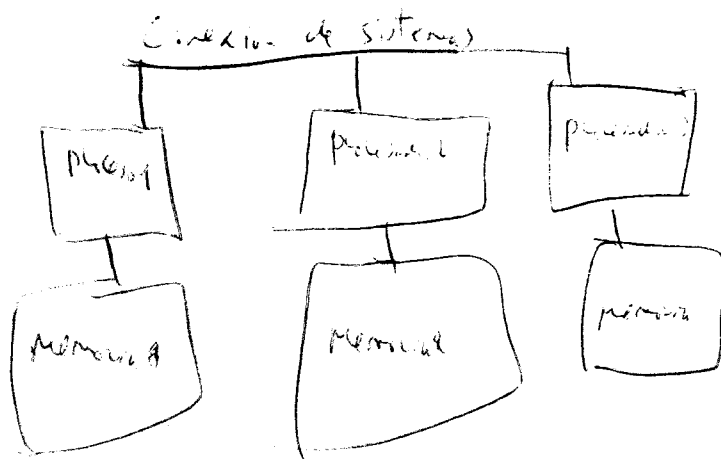
Cuando trabajas con varios procesadores, necesitas que trabajen con ~~entre~~ conjuntos (acoplados).

Cada procesador necesita ~~disponer~~ disponer de memoria RAM y 4 puertos de conexión de 2 formas:

- Procesador fuertemente acoplado: Hay una memoria común para todos los procesadores, que se comunican mediante un bus de conexión.



- Procesador débilmente acoplado: cada procesador asigna su propia parte de memoria. Se comunican mediante la línea de sistema.



~~La configuración también se puede~~

En multiprocesador se puede clasificar en:

• Simétrico: Todos los procesadores, disponen de la misma memoria.

• Asimétrico: Algunos procesadores, acceden a la memoria, pero habiendo en otros procesadores control.

## PROGRAMACIÓN CONCURRENTE

Cuando en un programa ocurren varios sucesos al mismo tiempo.

## PROGRAMACIÓN PARALELA

Cuando se programan de forma concurrente dentro de un mismo multiprocesador se denominan programación paralela. Cada procesador realiza un proceso al mismo tiempo.

Tengo acceso directo a los datos de otros aplicativos, o procesos.

## PROGRAMACIÓN DISTRIBUIDA

Cuando tengo varios equipos

Cuando dispongo de varios equipos en lugares geográficos diferentes que trabajan en conjunto al mismo tiempo; se llama programación distribuida.

Como no están dentro del mismo procesador, no hay acceso directo a la memoria ni variables del sistema.

Para evitar la redundancia, se necesita la comunicación. No se puede modificar directamente los variables, por lo que se usan réplicas de las variables en los distintos dispositivos para poder acceder.

Esta comunicación se realiza a través de redes de red (red local o internet).

La comunicación y la sincronización tienen que ser perfectas, porque si con un fallo falla la comunicación distribuida.

## VENTAJAS DE LA PROGRAMACIÓN CONCURRENTE

- Funcionamiento y ejecución más rápida y eficiente (Ventaja directa principal)
- Flexible: Permite distribuir los recursos para mejorar la eficiencia del sistema
- ~~Adaptabilidad~~
- Robustez a fallos: Si falla un proceso, el resto continúa excepto si depende del proceso que ha fallado.
- Especialización: Permite repartir los recursos para realizar un tarea ~~específica~~ específica.

### Definición

## CONVENCIONES DE LA PROGRAMACIÓN CONCURRENTE

- Necesidad de la correcta comunicación y sincronización, así como las comunicaciones en caso de fallo.
- Mayor complejidad en la hora de programar.

## PROCESOS, SEMÁFOROS E Hilos

Si un proceso trabaja en primer plano (foreground) puede interactuar con el usuario y se denomina proceso (programa). Si el proceso se ejecuta en segundo plano (background) se denomina servicio.

~~En los~~

Los servicios se ejecutan en segundo plano y sirven para dar soporte a los procesos (ya sea de forma directa o mediante ~~información~~ información).

Los procesos y servicios se denominan entidades pesadas (requieren mucho recurso y tiempo del procesador). Los hilos se denominan ~~entidades~~ entidades ligeras (requieren pocos recursos y duran poco tiempo).

Los hilos realizan acciones concretas que duran ~~constantemente~~ el tiempo que tardan en ejecutarse.

Los hilos realizan acciones concretas que solo están ~~dentro~~ dentro de los procesos el tiempo que tardan en ejecutarse.

Los hilos también se llaman ~~subprocesos~~ subprocesos, es decir que los procesos y los servicios se componen de varios hilos.

Un proceso a su vez en conjunto de hilos que se comunican entre ellos.  
Un programa de consola o ejecuta múltiples tareas al mismo tiempo.  
Si se interrumpe un hilo se interrumpe todo esto, pero se puede interrumper el proceso y se interrumpe todo los hilos.

## MCCOS

- Declara la clase herencia de Thread.
- Crea la función run() y llamada en start().
- Crea en constructor los atributos para los heredados de Thread o declarados.

## DETENER MCCOS

- Método stop();  
Destruye por completo el hilo incluido los recursos. Actualmente está obsoleto.
- Método interrupt();  
Interumpe el método, pero no lo para. Es necesario utilizar el método isInterrupted y manejar excepciones.

## PREGUNTA DORMIR

4 preguntas.

Infinitos intentos.

20 tiempo 20 segundos.

Si se acierta o se para a la siguiente.

Si se acaba el tiempo. BOOMB!