

Hilos

Monday, November 5, 2018

11:14 AM

► Procesos

- Secuencias / flujos de ejecución

► Hilo

- Es una secuencia o característica que permite a un sistema realizar varias tareas de forma simultánea

► Estados del proceso

- ① Nuevo → Se crea el proceso
- ② Ejecución → Cuando el CPU está ejecutando instrucciones
- ③ Listo → Cuando espera a ser asignado a la CPU
- ④ Bloqueado → Cuando espera o depende de otro proceso
- ⑤ Finalizado → Termina la ejecución

// Datos importantes

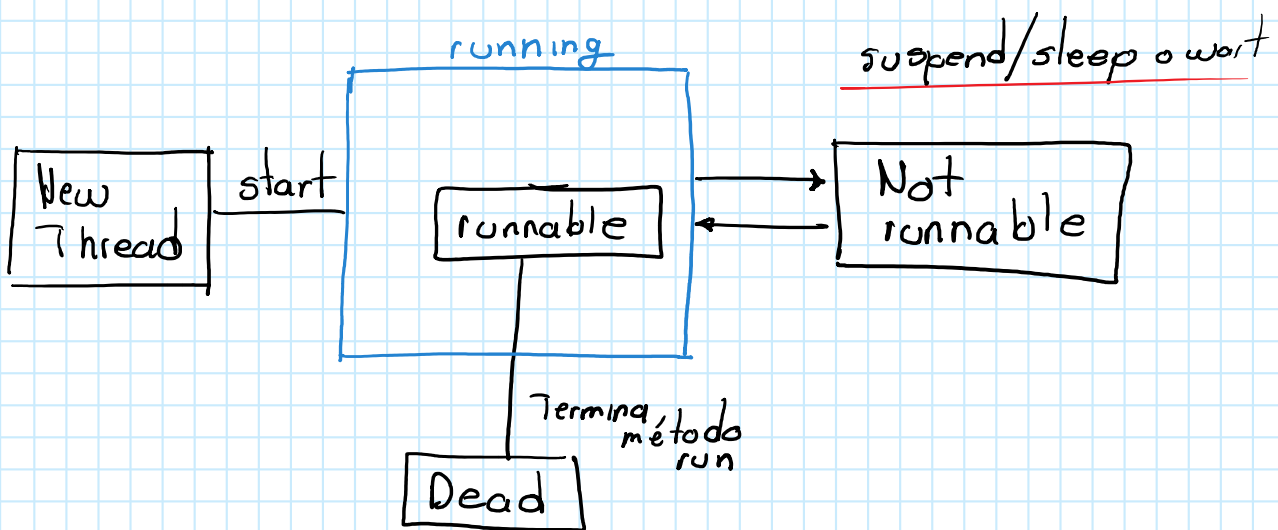
- Sistema Operativo asigna tiempo a cada uno de los procesos
- El tiempo que un proceso pasa en cada estado depende de su prioridad y de la política de scheduling.

de los procesos

- Los procesos se pueden bloquear entre si mismo.
- Los procesos pueden comunicarse entre si, compartiendo espacios de memoria.
- Un número excesivo de hilos puede llegar a degradar el rendimiento del sistema

→ La caída de un hilo puede matar la aplicación.

► Hilos - threads



► Uso de hilos

① Heredar de la clase "Thread"

② Método constructor
// Nombre genérico al proceso

```
public Hilo (String nombre) {
```

```
public Hilo (String nombre){  
    super(nombre);  
}
```

③ Sobreescibir run

```
public void run(){  
    //Lineas  
}
```

//Dentro del main

```
new Hilo ("Hilo 1").start();
```

👉 Interfaz Runnable

- Ya no se sobreescrituras solo se implementa
- class Hebra implements Runnable
- Se implementa el método write
→ Thread.currentThread().getName();


③ Implementación

- Instancia de un hilo mediante su interfaz
- new Thread (new Hebra(), "Hilo 1").start();

- `new Thread (new Hebra(), "Hilo 1").start();`

Grupo de hilos | `(ThreadGroup tg, String name)`

`ThreadGroup TG = new ThreadGroup ("Grupo :v")`

 Dar prioridad a un hilo

`.setMaxPriority (Thread.MAX_PRIORITY)`

Métodos Sincronizados

- Son aquellos que no se pueden acceder si un objeto está haciendo uso de ellos.
- "synchronized" palabra reservada
- `notifyAll` | Sirve para indicar cuando se ha terminado un proceso sincronizado
- `sleep` | Dormimos por un lapso de tiempo un proceso.

// La sincronía es para jerarquizar.

► Reusabilidad de Software

1) Patrones de arquitectura (Reusabilidad de estructuras)

- Soluciones al más alto nivel de software y hardware

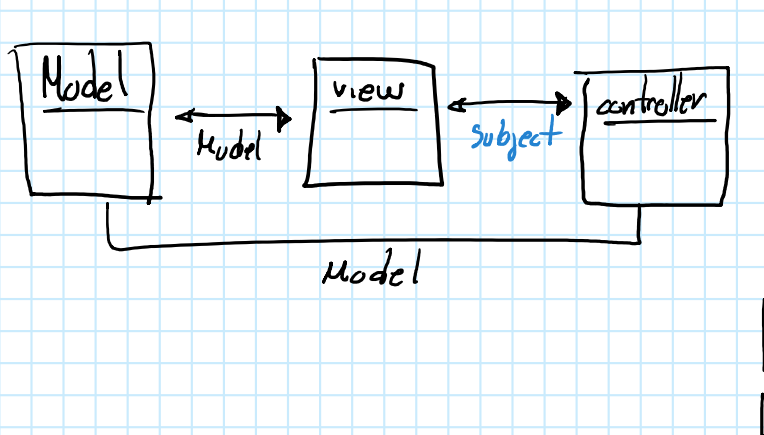
2) Patrones de diseño (Reusabilidad de diseño)

- Nivel intermedio de estructuras de software

3) Patrones de programación (Reusabilidad de código)

- Son al más bajo nivel, ya sea en clases y métodos.
- Son soluciones a nivel del lenguaje de programación

► MVC (Modelo Vista Controlador)



- MainFrame / Two-Tier Architecture / Three-Tier Architecture

▶ Patrones de diseño

- Es una descripción de clases y objetos comunicándose entre sí y adaptados para resolver un problema de diseño general

(Erich Gamma-Got)

▶ ¿Qué debe incluir?

- ① Nombre
 - ② Establece el problema
 - Aplicación
 - Propósito
 - ③ Solución
 - Estructura
 - Participantes
 - Colaboradores
- Quién ha utilizado el patrón ←

▶ Singleton

- Única en todo el sistema

▶ Consideraciones

- ① Nombre
- ② Descripción del problema
- ③ Solución
- ④ Consecuencias

▶ Patrones Verticales | Casos particulares

- Java Script
- Formulas Químicas

▶ Patrones horizontales | Aplicables a cualquier cosa

▶ Clasificación mediante propósito