

# Sincronización P2P mediante eventos y mediante barreras

Jesús Rodríguez Heras

1 de abril de 2019

- 1 Sincronización
- 2 P2P
- 3 Sincronización P2P mediante eventos
- 4 Sincronización mediante barreras
- 5 Bibliografía

# Sincronización

## Definición

Procedimiento por el que se ajusta el valor de un reloj a un tiempo físico de referencia con una precisión preestablecida.

## Importante

Es importante asegurar una correcta sincronización entre los computadores de un sistema distribuido para que no se produzcan derivas.

## Deriva

Representa el tiempo que un reloj se desvía respecto a la hora real con el paso del tiempo.

## Definición

Es un sistema auto-organizado de entidades iguales y autónomas cuyo objetivo es el uso compartido de recursos distribuidos en un ambiente de red evitando servicios centralizados.

## Ejemplo

BitTorrent, eMule,  $\mu$ Torrent, etc.

# Sincronización P2P mediante eventos

- Para simplificarlo, supondremos que tenemos dos procesadores conectados mediante una red P2P (pont-to-point).
- Para sincronizar dichos procesadores tendremos una estructura de espera activa con una variable (o flag) común en ambos procesadores.
- Cuando uno de ellos realiza un cambio sobre dicha variable compartida (evento) el segundo se sincroniza con el primero.

# Sincronización P2P mediante eventos

Ejemplo de dos procesos sincronizados mediante un evento donde inicialmente la variable global turno vale 0 (o un valor distinto de 1).

## Procesador 1

```
//Resto de código inicial
while(turno != 1){
    //Espera ocupada
}
//Código sincronizado
```

## Procesador 2

```
//Resto de código inicial
turno = 1;
//Código sincronizado
```

En este ejemplo podemos ver cómo el procesador 1 se queda a la espera de que el procesador 2 le pase el turno (estableciendo a 1 la variable turno). Con esto se sincronizan para realizar la sección de Código sincronizado o, simplemente, porque sea necesaria una sincronización en ese punto del programa.

## Barrera

Es un punto del código que ninguna entidad concurrente sobrepasa hasta que todas las entidades existentes han llegado a ella.

# Sincronización mediante barreras

Ejemplo de código Java para la gestión de una barrera y varios procesos:

```
//Código de Thread:
public Hilo extends Thread{
    public Hilo (CyclicBarrier ba) {...}
    public void run(){
        try{
            int i = bar.await();
        }catch(Exception e){}
        //Código a ejecutar cuando se abre barrera...
    }
}

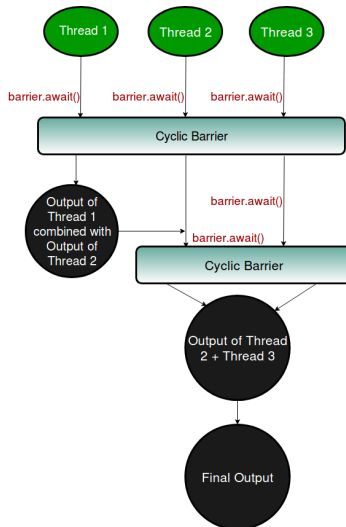
//Programa principal:
int numHilos = n; //Número de hilos que abren barrera
CyclicBarrier Barrera = new CyclicBarrier(numHilos);
new Hilo(Barrera).start();
```



# Sincronización mediante barreras

En el ejemplo anterior se ve cómo, hasta que no llegan todos los procesos a la barrera, el programa no seguirá ejecutándose. Y, cuando lo haga, lo harán todos los procesos a la vez.

A la derecha tenemos una figura donde se lanzan tres procesos a la vez y son sincronizados de por medio.





## Sincronización de eventos discretos

*<http://sedici.unlp.edu.ar/handle/10915/23619>*



## Sincronización de eventos discretos en sistemas distribuidos

*[http://sedici.unlp.edu.ar/bitstream/handle/10915/23619/4787-paper\\_cacic\\_2012\\_a\\_presentar.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/23619/4787-paper_cacic_2012_a_presentar.pdf?sequence=1)*



## Sincronización por barrera

*<http://webdiis.unizar.es/~ezpeleta/lib/exe/fetch.php?media=misdatos:pc:08.pdf>*



## Sincronización por barrera, Grupo ARCOS

*<http://ocw.uc3m.es/ingenieria-informatica/arquitectura-de-computadores/materiales/es-m5-03-sincr-ocw.pdf>*