

Practica 4: Relojes

Computación Distribuida 2021-1

Profesora: Karla Vargas

Diego Estrada

Jorge Cortés

Ricardo Desales

José Morán

22 de noviembre de 2020

1. Introducción

Esta práctica servirá para que comprendan el concepto de reloj en un sistema distribuido asíncrono. Implementarán dos relojes (Lamport y vectorial) *encima* de algoritmos anteriormente vistos.

2. Desarrollo

Para esta práctica convertiremos nuestro sistema síncrono en parcialmente asíncrono. Esto lo lograremos esperando tiempos aleatorios entre cada envío y recepción de mensajes (observen que como estos valores están acotados, nuestro sistema no será completamente asíncrono).

Para implementar el reloj de Lamport usaremos como *base* el algoritmo de Broadcast. Tendrán que usar el mismo archivo que entregaron para esa práctica, pero adaptandolo para que el sistema sea asíncrono y que cada nodo tenga una variable `reloj`.

Para el vectorial usarán el algoritmo de DFS como base. Harán lo mismo que para el de Lamport, pero en vez de que su reloj sea un entero, será un arreglo de éstos (Nota: Tendrán que modificar su constructor para decirle al nodo cuantos nodos hay en la gráfica)

En ambos algoritmos, aparte de eso, tendrán que guardar, en una lista, todos los eventos que hayan ocurrido con ese nodo (envío y recepción de mensajes). Cada elemento de esta lista de eventos deberá tener el valor del reloj que tenía el nodo al enviar (o recibir) el mensaje, el tipo de evento (Usen la cadena E cuando se trate de envío, R cuando se trate de recibir) el mensaje, el nodo que envió y el nodo que recibió (en ese orden).

3. Pruebas

Para esta práctica no les daremos un *esqueleto* pues usarán algoritmos ya hechos. Les daremos el archivo `test.py` (se los enviaremos a más tardar el martes), lo tendrán que poner en la carpeta donde están los archivos de sus nodos y hacer:

```
myUser $> pytest -q test.py
```

Estas pruebas son distintas, ya que vamos a revisar que los eventos que pasaron en la ejecución respeten el orden causal.

4. Observaciones

- Recuerden que lo importante de los relojes es verificar que éste le asigna valores temporales correctos a cada evento.

- Para los números aleatorios pueden usar la biblioteca `random` de python.