Universidad Nacional Autónoma de México

Facultad de Ciencias

Computación Distribuida

Tarea 6

Johann Ramón Gordillo Guzmán
418046090

Tarea presentada como parte del curso de **Computación Distribuida** impartido por la profesora **M.C Karla Rocío Vargas Godoy**.

17 de Noviembre del 2020

Link al código fuente: https://github.com/JohannGordillo/

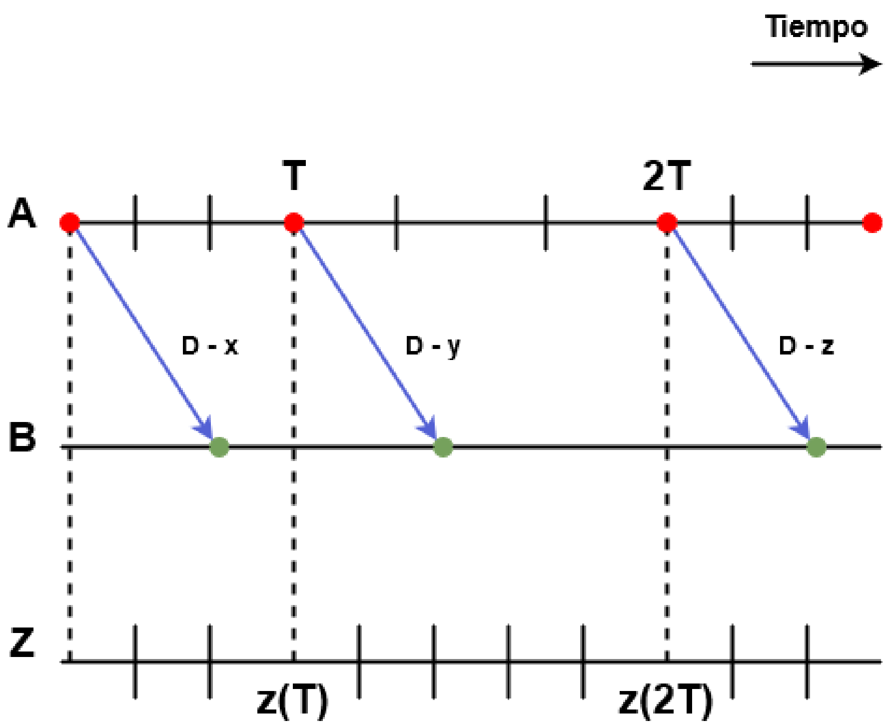
Actividades

1. Sean A y B dos procesos cuyos relojes no están sincronizados pero ambos tienen drift acotado. Cuando A manda un mensaje a B, el tiempo máximo real que tarda en llegar el mensaje es a lo más D. Supón que tienes un algoritmo en el que A manda mensajes cada T unidades de su tiempo. Para ir midiendo las unidades de tiempo, A tiene un contador que aumenta cada tanto (evento de cómputo local).

   - Dibuja una ejecución α donde A manda 3 mensajes a B, en donde por lo menos uno de estos se tarda el tiempo máximo permitido.

     Respuesta.

     Primero que nada, supongamos que $D \leq z(T)$ (esto es válido, lo vimos en la última clase). Consideremos ahora la siguiente ejecución α:



     Donde los retardos en los envíos de mensajes son $0 \leq D - x, D - y, D - z \leq D$, y además x, y o z es cero, pues al menos un envío tarda D. La tercera línea corresponde al reloj de referencia.

- De la ejecución α haz una lista de eventos, ordenandolos acorde a su causalidad.

  Respuesta.

  Denotemos con ϕ_1 el primer envío de un mensaje del proceso A al proceso B; con ϕ_2, la recepción del primer mensaje por B; con ϕ_3, el segundo envío de un mensaje; con ϕ_4 la segunda recepción; con ϕ_5 el último envío, y con ϕ_6 la última recepción.

  Para los eventos dentro de un mismo proceso tenemos:

  - En A.

    Primer envío con el segundo envío:
    $$\phi_1 \overset{\alpha}{\Longrightarrow} \phi_3$$

    Segundo envío con el tercer envío:
    $$\phi_3 \overset{\alpha}{\Longrightarrow} \phi_5$$

  - En B.

    Primera recepción con la segunda recepción:
    $$\phi_2 \overset{\alpha}{\Longrightarrow} \phi_4$$

    Segunda recepción con la tercera recepción:
    $$\phi_4 \overset{\alpha}{\Longrightarrow} \phi_6$$

  Para los eventos en distintos procesos:

  Primer envío con primera recepción:
  $$\phi_1 \overset{\alpha}{\Longrightarrow} \phi_2$$

  Segundo envío con segunda recepción:
  $$\phi_3 \overset{\alpha}{\Longrightarrow} \phi_4$$

  Tercer envío con tercera recepción:
  $$\phi_5 \overset{\alpha}{\Longrightarrow} \phi_6$$

  Finalmente, usando transitividad:

Para el evento ϕ_1, tenemos:

$$\phi_1 \overset{\alpha}{\Longrightarrow} \phi_2$$

$$\phi_1 \overset{\alpha}{\Longrightarrow} \phi_3$$

$$\phi_1 \overset{\alpha}{\Longrightarrow} \phi_4$$

$$\phi_1 \overset{\alpha}{\Longrightarrow} \phi_5$$

$$\phi_1 \overset{\alpha}{\Longrightarrow} \phi_6$$

Para el evento ϕ_2, tenemos:

$$\phi_2 \overset{\alpha}{\Longrightarrow} \phi_4$$

$$\phi_2 \overset{\alpha}{\Longrightarrow} \phi_6$$

Para el evento ϕ_3, tenemos:

$$\phi_3 \overset{\alpha}{\Longrightarrow} \phi_4$$

$$\phi_3 \overset{\alpha}{\Longrightarrow} \phi_5$$

$$\phi_3 \overset{\alpha}{\Longrightarrow} \phi_6$$

Para el evento ϕ_4, tenemos:

$$\phi_4 \overset{\alpha}{\Longrightarrow} \phi_6$$

Para el evento ϕ_5, tenemos:

$$\phi_5 \overset{\alpha}{\Longrightarrow} \phi_6$$

De esta manera, podemos ordenar los eventos de la ejecución α de acuerdo a su causalidad como sigue:

$$\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6$$

- ¿Cuánto tiempo puede tardar a lo más en terminar α? ¿Y el mínimo?

  Respuesta.

  Para hablar de tiempo mínimo y de tiempo máximo, tenemos que fijarnos en el reloj de referencia.

  El último mensaje se manda en el tiempo $2T$ del proceso A, sin embargo este tiempo es $z(2T)$ en el reloj de referencia. Por lo que, considerando que el último mensaje tarde D unidades de tiempo en llegar, el tiempo máximo de la ejecución α será $z(2T) + D$.

  Por otro lado, el tiempo mínimo es $z(2T) + 1$, ya que es el caso en el que $D = 1$.

  Creo que con el argumento anterior basta para responder a la pregunta.

  Sin embargo, podemos ponernos un poco más formales. El drift de un reloj k en el instante del tick i se representa por:
  $$drift_i^k = \frac{z(tick_i^k) - z(tick_{i+1}^k)}{n^k}$$
  donde n^k es la granularidad del reloj k y z la marca de tiempo absoluta.

  Denotemos por A al reloj de procesador A y supongamos que su granularidad es n^A. Sabemos que el drift del reloj de A está acotado. Es decir, existe $m \in \mathbb{Z}$ tal que:

  $$drift_i^A = \frac{z(tick_i^A) - z(tick_{i+1}^A)}{n^A} \leq m$$

  en el instante del tick i.

  Por lo anterior, tenemos:
  $$z(tick_i^A) - z(tick_{i+1}^A) \leq mn^A$$
  y con base en esto podemos proceder a hacer el análisis de los tiempos de la ejecución α.

  Ahora, en el peor de los casos:

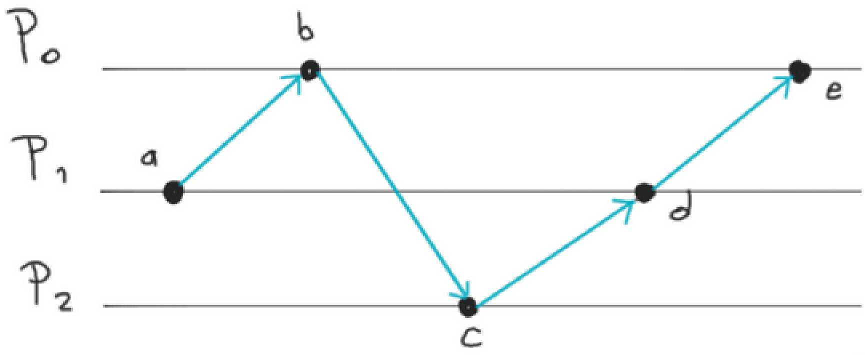
  $$z(tick_i^A) - z(tick_{i+1}^A) = mn^A$$

  Es decir, en el peor de los casos cada tick de A durará mn^A ticks del reloj de referencia.

  Tras $2T$ ticks del reloj de A, en el reloj de referencia habrán pasado $2Tmn^A$, por lo que el tiempo máximo que puede tardar la ejecución α es $2Tmn^A + D$.

  Por otro lado, sabemos que el reloj de A no puede ir más rápido que el reloj de referencia, por lo que en el mejor de los casos cada tick en A va a equivaler a 1 tick en el reloj de referencia. Entonces, tras $2T$ ticks de A tendremos $2T$ ticks en el reloj de referencia, por lo que el tiempo mínimo de ejecución de α será $2T + 1$, pues en el mejor de los casos $D = 1$.

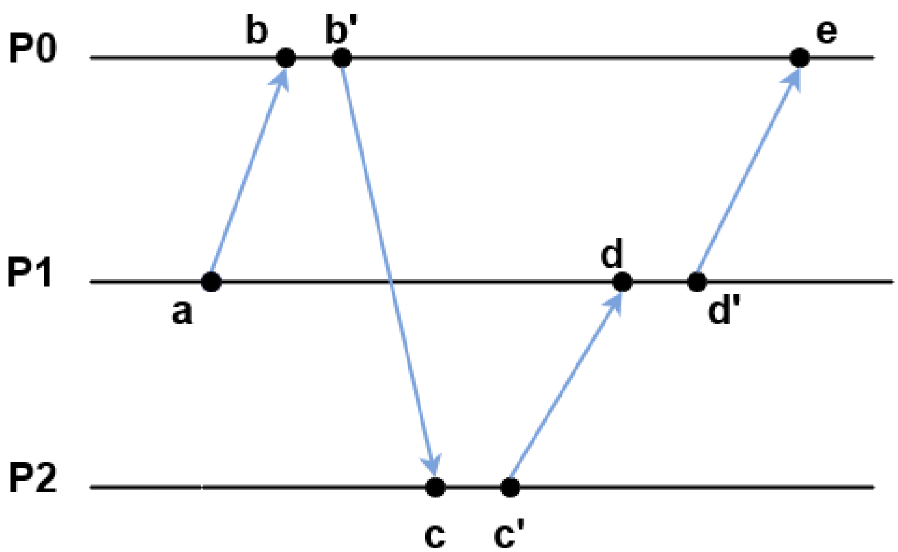
2. Considera la ejecución α de la Figura x. Los retardos máximos para cada uno de los eventos son los siguientes: $a \to b$ es $B_{a,b}$, $b \to c$ es $B_{b,c}$, $c \to d$ es $B_{c,d}$, $d \to e$ es $B_{d,e}$, $a \to e$ es $B_{a,e}$ y $b \to d$ es $B_{b,d}$. Los retardos mínimos son similares, pero invirtiendo el orden de las letras en B, i.e. el mínimo tiempo que puede tomar en pasar $a \to b$ es $B_{b,a}$.



- ¿Cuánto vale el retardo máximo de b a d en términos de las B's? Es decir, dado $z(d) - z(b) \leq x$, ¿Cuánto vale x?

  Respuesta.

  Haciendo la corrección de la figura:



  El retardo máximo de b a d, denotado como $B_{b,d}$, es la suma de los retardos máximos de b' a c y de c' a d.

  Es decir, el retardo máximo es $B_{b,d} = B_{b',c} + B_{c',d}$. De esta manera, $z(d) - z(b) \leq B_{b',c} + B_{c',d}$.

- ¿Cuánto vale el retardo mínimo de toda la ejecución? ¿Y el máximo?

  Respuesta.

  El retardo máximo es $B_{a,e}$, que se calcula como:

$$B_{a,e} = B_{a,b} + B_{b',c} + B_{c',d} + B_{d',e}$$

  El retardo mínimo es $B_{e,a}$, que se calcula como:

$$B_{e,a} = B_{e,d'} + B_{d,c'} + B_{c,b'} + B_{b,a}$$

  Así, $B_{a,e}$ y $B_{e,a}$ son los retardo máximo y mínimo, respectivamente.

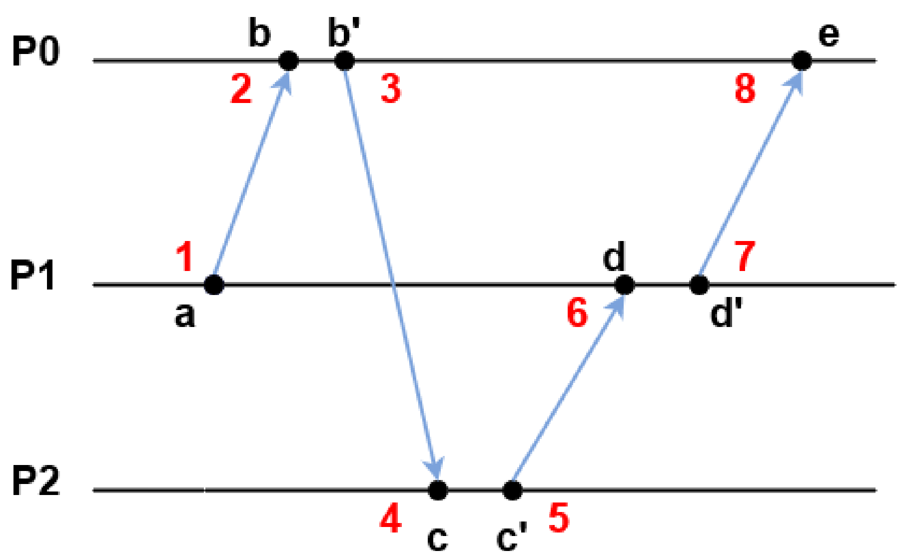
- Anotar los eventos con los relojes lógicos y los relojes vectoriales.

  Respuesta.

  - **Relojes Lógicos.**

    Todos los procesos van a tener un contador que inicialmente comienza en 1.

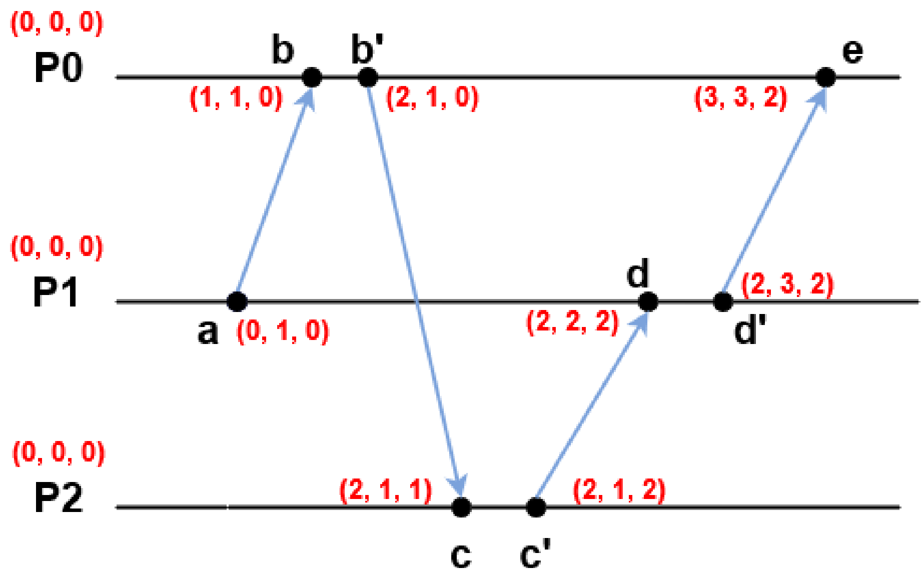
    Gráficamente se ve como sigue:



    Notemos que un proceso incrementa su contador antes de cada evento en un proceso. Además, cuando un evento manda un mensaje, éste incluye su contador en él, y cuando otro proceso recibe el mensaje, cambia su contador para quedarse con el máximo valor entre su valor actual y el recibido antes de considerar el mensaje recibido.

- **Relojes Vectoriales.**

  Gráficamente se ve como sigue:



  Inicialmente todos los relojes vectoriales son cero. Los vectores son de tamaño 3 y cada vez que un proceso lleve a cabo un cómputo local, éste incrementa su entrada en 1. Cuando un proceso manda un mensaje, manda al vector junto con el mensaje. Cuando un proceso recibe un mensaje, incrementa su entrada en el vector en 1m y actualiza cada elemento en su vector tomando el máximo de cada entrada entre su vector y el vector recibido.

3. Da un ejemplo de una situación de la vida real con relación de causalidad.

   Respuesta.

   Supongamos los siguientes eventos de la vida cotidiana:
   ϕ_1: Alguien enciende un cigarrillo.
   ϕ_2: Ocurre una explosión.

   Si ϕ_1 ocurre después de ϕ_2 es poco probable que exista un orden causal. No suena lógico que después de una explosión alguien decida encender un cigarrillo.

   Si ϕ_2 ocurre después de ϕ_1, puede haber un orden causal, ya que la persona al encender el cigarrillo pudo haber ocasionado la explosión si había una fuga de gas.

Referencias

[1] Raynal, M. *Distributed Algorithms for Message-Passing Systems*. Springer, 2013.

[2] Aspnes, J. *Notes on Theory of Distributed Systems*. Yale University, 2017.