

Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad
- coordinación
- transacciones

tiempo y estados globales

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

- ⌚ Monitorización y temporización de ejecuciones
- ⌚ Necesidad de conocer cuando ocurrió un evento
- ⌚ Algoritmos de sincronización
 - Manteniendo de consistencia en transacciones
 - Protocolos de autenticación
 - ...
- ⌚ ¿Existe un reloj universal de referencia?
 - Teoría Especial de la Relatividad de Einstein
 - Causa física y efecto físico
 - Temporización de la causa y el efecto
 - Tiempo físico absoluto de Newton



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

❏ Noción de tiempo es también problemática en un sistema distribuido (SD):

- No existe un reloj global al sistema

❏ Aproximaciones:

- Sincronización de los relojes de los computadores
- Relojes lógicos, vectoriales, ...
- Algoritmos para conocer los estados globales de un SD cuando ejecuta



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

- ⌚ Cada computador de la red tiene su propio reloj interno:
 - usado por los procesos locales para obtener el valor del tiempo actual
 - procesos en computadores distintos pueden tener marcas de tiempo distintas
 - los relojes derivan con respecto al tiempo perfecto y las tasas de deriva también difieren entre ellos
- ⌚ Aunque todos los relojes del SD se sincronicen, estos variarán significativamente con el tiempo



tiempo y estados globales

relojes, eventos y estados

Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones



Cada computador de la red tiene su propio reloj

Tasa de deriva de reloj: diferencia por unidad de tiempo en que el reloj del computador difiere del reloj perfecto

- procesos en computadores distintos pueden tener

¿Cuáles son las **necesidades del *make de Unix*** con respecto al reloj, en un sistema NFS?



Aunque todos los relojes del SD se sincronicen, estos variarán significativamente con el tiempo



Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

- ② Un SD está definido como una colección P de N procesos $p_i, i = 1, 2, \dots, N$
- ② Cada proceso p_i tiene un estado s_i formado por todas sus variables u objetos y que puede cambiar en ejecución
- ② Se comunican a través de la red mediante mensajes
- ② Las acciones que puede realizar un proceso:
 - *Enviar, Recibir, Cambiar estado*
- ② **Evento**: ocurrencia de una acción que lleva a cabo un proceso al ejecutar, p.e. Enviar, recibir, ...
- ② Los eventos en el proceso p_i , pueden ordenarse de forma total por la relación \rightarrow_i "suceder antes en p_i "
- ② **Historia del proceso p_i** es una serie de sus eventos ordenados por \rightarrow_i : $historia(p_i) = h_i = \langle e_i^0, e_i^1, e_i^2, \dots \rangle$



Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

Relojes:

- ⌚ Para establecer las marcas temporales se usa el reloj del computador
- ⌚ En un instante t el SO lee el valor del reloj hardware del computador : $H_i(t)$
- ⌚ Calcula el tiempo en el reloj software:
$$C_i(t) = \alpha H_i(t) + \beta \text{ (escala y compensa)}$$
 - p.e. un número de 64-bit dando los nanosegundos desde un tiempo base
 - En general no es completamente exacto
 - Pero si C_i se comporta suficientemente bien, puede ser usado como marcador de los eventos de p_i



Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

Relojes:

- Para establecer las marcas temporales se usa el reloj del computador

- En un instante t el SO lee el valor del reloj hardware del computador: $H(t)$
- Calcula el tiempo en el reloj software:

$$C(t) = \alpha H(t) + \beta \text{ (escalado y compensación)}$$

Resolución del reloj: periodo entre dos actualizaciones consecutivas del reloj

- En general no es completamente exacto

- Pero si C_i se comporta suficientemente bien, puede ser usado como marcador de los eventos de p_i



tiempo y estados globales

relojes, eventos y estados

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones



Red

- Los relojes no siempre están en perfecto acuerdo
- Sesgo:** diferencia de tiempo entre dos relojes en un instante determinado
- Tasa de deriva:**
 - Los relojes de cuarzo ordinarios derivan 1 seg. cada 11-12 días (10^{-6} segs/seg)
 - Los relojes de alta precisión derivan 10^{-7} ó 10^{-8} segs/seg



tiempo y estados globales

relojes, eventos y estados

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones



Red

Los relojes no siempre están en perfecto acuerdo
¿qué puede influir en la **tasa de**
desviación del reloj?
un instante determinado

Tasa de deriva:

- Los relojes de cuarzo ordinarios derivan 1 seg. cada 11-12 días (10^{-6} segs/seg)
- Los relojes de alta precisión derivan 10^{-7} ó 10^{-8} segs/seg



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Tiempo Universal Coordinado (UTC)

- ⌚ Los relojes se pueden sincronizar con fuentes externas muy precisas: Tiempo atómico Internacional basado en un reloj con deriva de 10^{-13} (un seg. cada 300.000 años)
- ⌚ UTC es un estándar internacional de establecimiento y mantenimiento del tiempo transcurrido
- ⌚ Está basado en el tiempo atómico y ocasionalmente ajustado al tiempo astronómico
- ⌚ La señal se difunde mediante estaciones de radio por tierra y mediante satélites. Las computadoras pueden sincronizar sus relojes mediante receptores adecuados
 - estaciones terrestres tienen una precisión entre 0.1-10 miliseg.
 - GPS tiene una precisión de 1 microseg.



Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

Tiempo Universal Coordinado (UTC)

- Los relojes se pueden sincronizar con fuentes externas muy precisas: Tiempo atómico Internacional basado en un reloj con deriva de 10^{-13} (un seg. cada 300.000 años)
- UTC es un estándar internacional de establecimiento y mantenimiento del tiempo transcurrido
- ¿Por qué no poner **receptores GPS** en todos los computadores?
- Está basado en el tiempo atómico y ocasionalmente ajustado al tiempo astronómico
- La señal se difunde mediante estaciones de radio por tierra y mediante satélites. Las computadoras pueden sincronizar sus relojes mediante receptores adecuados
 - estaciones terrestres tienen una precisión entre 0.1-10 miliseg.
 - GPS tiene una precisión de 1 microseg.



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

- ④ **Sincronización externa:** un reloj C_i se sincroniza con una fuente UTC exacta S :
 - $|S(t) - C_i(t)| < D$ para $i = 1, 2, \dots, N$ en un intervalo I de tiempo real
 - Los relojes C_i son precisos con el límite D .
- ④ **Sincronización interna:** cualquier par de computadores están sincronizados si sus relojes cumplen:
 - $|C_i(t) - C_j(t)| < D$ para $i, j = 1, 2, \dots, N; i \neq j$ en un intervalo I de tiempo
 - Los relojes C_i y C_j concuerdan con el límite D .
- ④ Relojes sincronizados internamente no necesariamente lo están externamente, puesto que pueden derivar juntos
- ④ Si el conjunto P está sincronizado externamente con un límite D , ¿también lo está internamente?



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Corrección del reloj:

- ⌚ Se dice que un reloj hardware (H) es correcto si su límite de deriva es conocido $\rho > 0$. (p.e. 10^{-6} secs/ sec)
- ⌚ Por tanto, el error en la medida de dos eventos en t y t' está limitado:
 - $(1 - \rho)(t' - t) \leq H(t') - H(t) \leq (1 + \rho)(t' - t)$ donde $t' > t$
 - Esto impide que se produzcan saltos traumáticos en el valor leído
- ⌚ Se puede relajar la condición (monotonicidad)
 - $t' > t \Rightarrow C(t') > C(t)$
 - p.e. requerido por el *make* de Unix
 - se puede alcanzar la monotonicidad en un reloj hardware que funciona rápido, ajustando los valores de α y β en $C_i(t) = \alpha H_i(t) + \beta$
- ⌚ Un **reloj defectuoso** es aquel que no cumple ninguna de las condiciones de corrección
- ⌚ Un **fallo de ruptura** de reloj: el reloj se para, no emite tics
- ⌚ Un **fallo arbitrario**: cualquier otro fallo...



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Corrección del reloj:

- ⌚ Se dice que un reloj hardware (H) es correcto si su límite de deriva es conocido $\rho > 0$. (p.e. 10^{-6} secs/ sec)
- ⌚ Por tanto, el error en la medida de dos eventos en t y t' está limitado:
 - $(1 - \rho)(t' - t) \leq H(t') - H(t) \leq (1 + \rho)(t' - t)$ donde $t' > t$

- ⌚ **Importante:** un reloj no tiene por qué ser preciso para ser correcto
 - Esto impide que se produzcan saltos traumáticos en el valor leído
 - p.e. requerido por el make de Unix

- se puede alcanzar la monotonicidad en un reloj hardware que funciona rápido, ajustando los valores de α y β en $C_i(t) = \alpha H_i(t) + \beta$
- ⌚ Un **reloj defectuoso** es aquel que no cumple ninguna de las condiciones de corrección
- ⌚ Un **fallo de ruptura** de reloj: el reloj se para, no emite tics
- ⌚ Un **fallo arbitrario**: cualquier otro fallo...



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Sincronización en un SD síncrono:

- @ Se dice que un SD es síncrono si están definidos los límites siguientes:
 - Tiempo máximo y mínimo para ejecutar cada paso de un proceso
 - Tiempo máximo y mínimo de recepción de un mensaje
 - Los límites de deriva de cada reloj local donde se ejecuta cada proceso son conocidos

Primera aproximación (sincronización interna):

- @ Un proceso p_1 envía su tiempo local t al proceso p_2 en un mensaje m ,
 - @ p_2 podría poner su reloj a $t + T_{trans}$, donde T_{trans} es el tiempo de transmisión de m ,
 - @ T_{trans} es desconocido, pero $min \leq T_{trans} \leq max$
 - @ La incertidumbre $u = max - min$. Si se establece el reloj a $t + (max - min)/2$ entonces el sesgo $\leq u/2$
-

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Sincronización en un SD síncrono:

- Se dice que un SD es síncrono si están definidos los límites siguientes:
 - Tiempo máximo y mínimo para ejecutar cada paso de un proceso
 - Tiempo máximo y mínimo de recepción de un mensaje
 - Los límites de deriva de cada reloj local donde se ejecuta cada proceso son conocidos

Primera aproximación (sincronización interna):

¿Es **Internet** un sistema síncrono?

- Un proceso p_1 envía su tiempo local t al proceso p_2 en un mensaje m ,
- p_2 podría poner su reloj a $t + T_{trans}$, donde T_{trans} es el tiempo de transmisión de m ,
- T_{trans} es desconocido, pero $min \leq T_{trans} \leq max$
- La incertidumbre $u = max - min$. Si se establece el reloj a $t + (max - min)/2$ entonces el sesgo $\leq u/2$

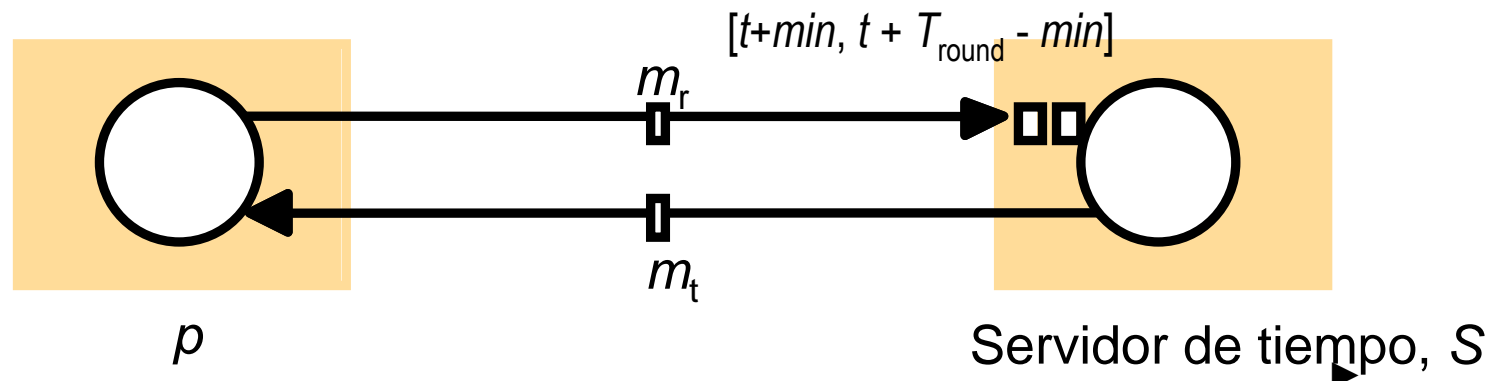
Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Método de Cristian (para sincronizar relojes externamente):

Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S
- p establece su tiempo a $t + T_{\text{round}}/2$ [T_{round} es el tiempo de ida y vuelta]
- Precisión: $\pm (T_{\text{round}}/2 - \text{min})$ [min es el mínimo estimado de transm.]
 - el momento más temprano en que S pone t en m_t es min después de que p enviara m_r
 - el momento más tardío es min antes de que m_t llegue a p
 - el tiempo de S cuando m_t llega está en el rango:



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Método de Cristian (para sincronizar relojes externamente):

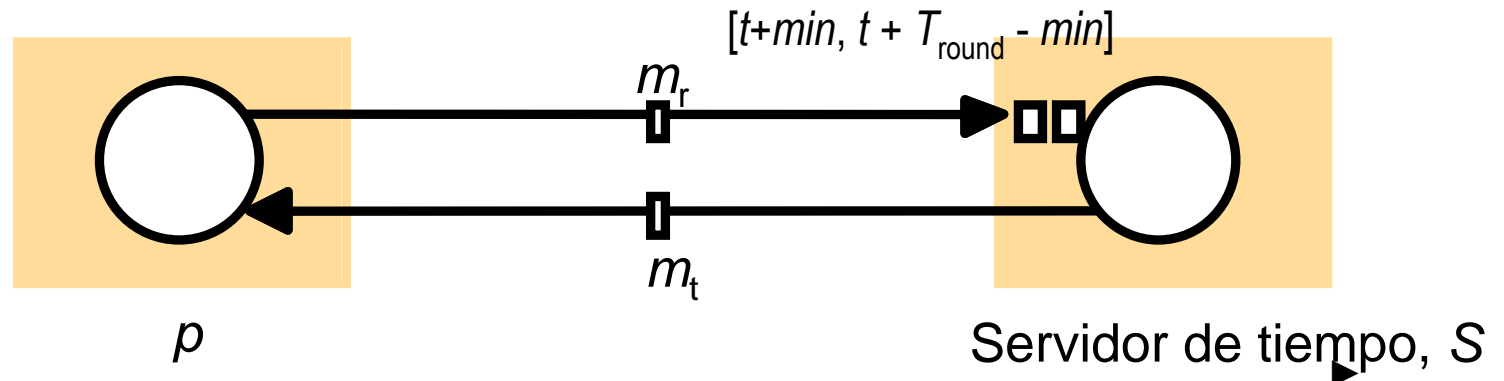


Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S

¿Potenciales **problemas** de este algoritmo?

- el momento más temprano en que S pone t en m_t es min después de que p enviara m_r
- el momento más tardío es min antes de que m_t llegue a p
- el tiempo de S cuando m_t llega está en el rango:



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Algoritmo de Berkeley (para sincronizar relojes internamente):

- ⌚ Un *maestro* consulta y recoge valores de reloj del resto de computadores, *esclavos*.
- ⌚ El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos.
- ⌚ Promedia los resultados incluyéndose y eliminando cualquier valor que no sea consistente.
- ⌚ Envía la magnitud de ajuste de cada reloj, puede ser positivo o negativo.
- ⌚ Experimentos:
 - 15 computadoras, tiempo de sincronización 20-25 millisegs. Tasa de deriva de relojes locales $< 2 \times 10^{-5}$
 - Si el maestro falla, se puede elegir a un nuevo maestro.



tiempo y estados globales

sincronización de relojes físicos

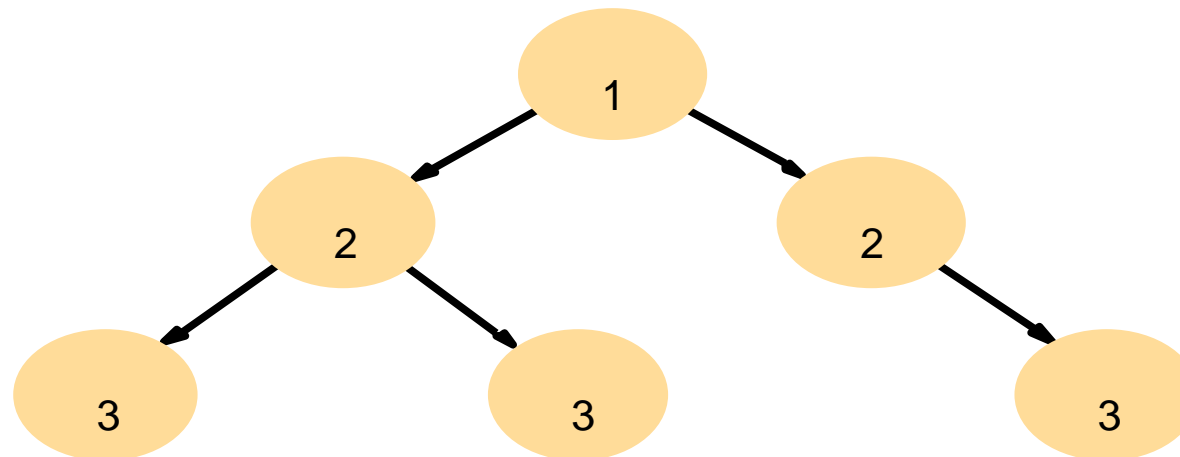
Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Protocolo de Tiempo de Red (NTP):

⌚ Servicio de tiempo para Internet. Sincroniza a los clientes con UTC

Servicio fiable, redundante, reconfigurable si alguno cae, escalable, con autenticación de las fuentes de tiempo



tiempo y estados globales

sincronización de relojes físicos

Contenido

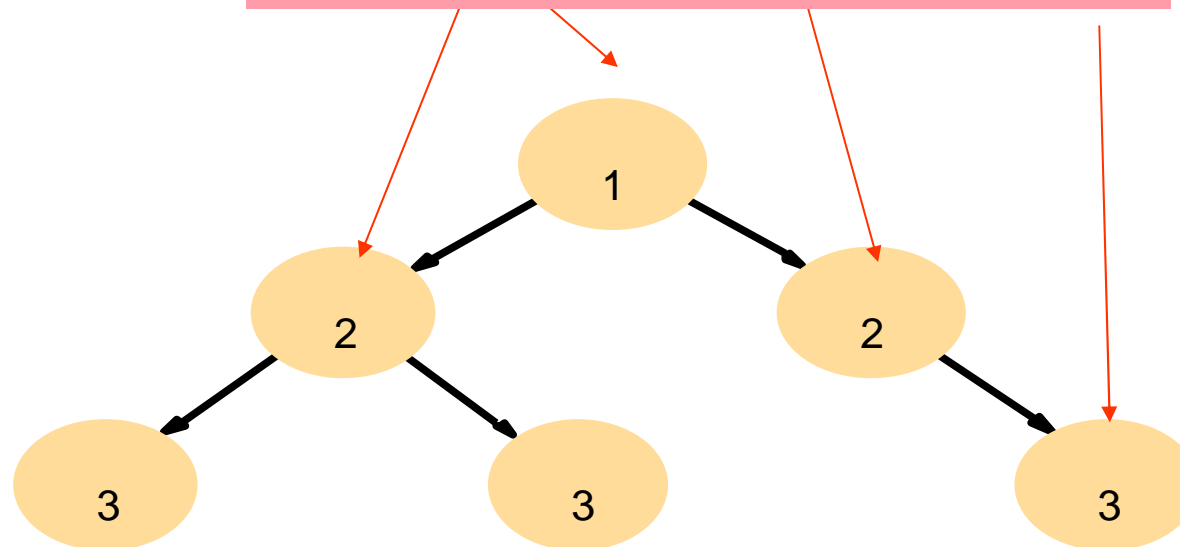
- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad
- coordinación
- transacciones

Protocolo de Tiempo de Red (NTP):

Ⓢ Servicio de tiempo para Internet. Sincroniza a los clientes con UTC. Los servidores primarios están conectados a fuentes UTC.

Servidores secundarios sincronizados a los primarios.

[Subred de sincronización] y en el nivel más bajo de servidores están los PC.



Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

NTP. Sincronización de servidores:

- ⌚ La subred de sincronización se puede reconfigurar si se produce un fallo:
 - un primario que pierde su conexión con UTC puede pasar a secundario
 - un secundario que pierde a su primario puede elegir otro primario

⌚ Modos de sincronización:

- Multidifusión (multicast)
 - En LAN de alta velocidad. Un servidor reparte el tiempo al resto que establecen su tiempo asumiendo un retraso de transmisión (no preciso)
- Llamada a procedimiento
 - Similar a de Cristian. El servidor acepta peticiones. Precisión más alta
- Simétrica
 - Pares de servidores se intercambian mensajes conteniendo información de tiempo
 - Usado en los casos en que se necesita muy alta precisión (p.e. en primeros niveles)



tiempo y estados globales

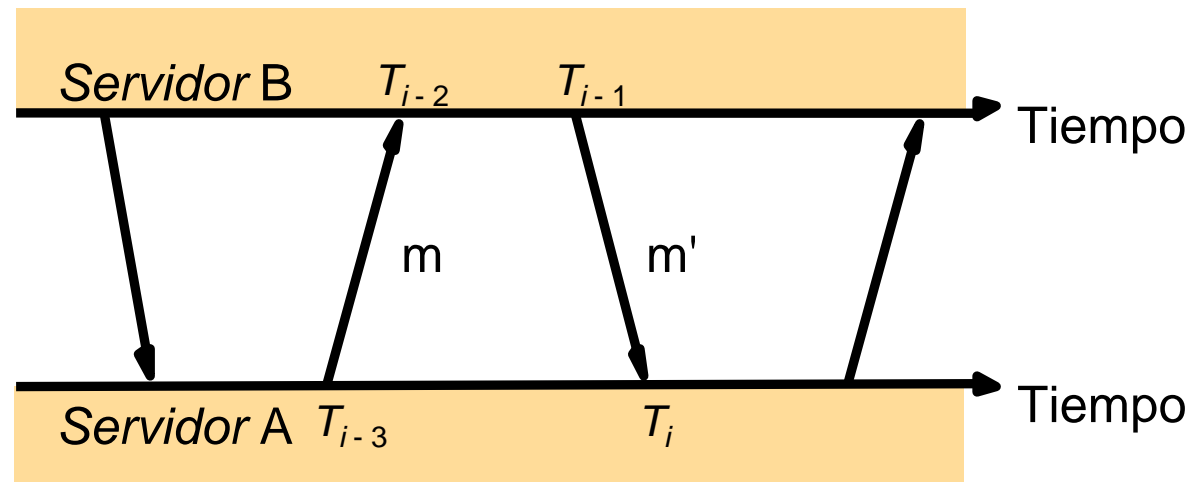
sincronización de relojes físicos

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

NTP. Intercambio de mensajes entre pares de servidores:

- Ⓢ Todos los modos usan UDP
- Ⓢ Cada mensaje lleva marcas de tiempo de los eventos recientes:
 - Tiempos locales de Envío y Recepción del mensaje anterior m
 - Tiempo local de envío del mensaje actual m'
- Ⓢ El receptor anota el tiempo local cuando recibe, T_i (tenemos $T_{i-3}, T_{i-2}, T_{i-1}, T_i$)
- Ⓢ Puede haber un retraso entre la llegada de un mensaje y el envío del siguiente y se pueden perder mensajes...



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

NTP. Precisión:

- Para cada par de mensajes entre servidores, NTP estima una compensación o , entre los dos relojes y un retardo d_i (tiempo total de transmisión para los dos mensajes t y t')

$$T_{i-2} = T_{i-3} + t + o \text{ y } T_i = T_{i-1} + t' - o$$

- Sumando las ecuaciones:

$$d_i = t + t' = T_{i-2} - T_{i-3} + T_i - T_{i-1}$$

- Restando las ecuaciones:

$$o = o_i + (t' - t)/2 \text{ donde } o_i = (T_{i-2} - T_{i-3} - T_i + T_{i-1})/2$$

- Como $t, t' > 0$ se puede ver que

$$o_i - d_i/2 \leq o \leq o_i + d_i/2 .$$

- por tanto o_i es una estimación de la deriva y d_i es una medida de la precisión

- Los servidores NTP mantienen pares del tipo $\langle o_i, d_i \rangle$, estimando la fiabilidad de las variaciones y permitiendo cambiar el propio par

- Ej.: precisión del orden de decenas de mseg. sobre Internet y de 1 mseg. sobre LAN



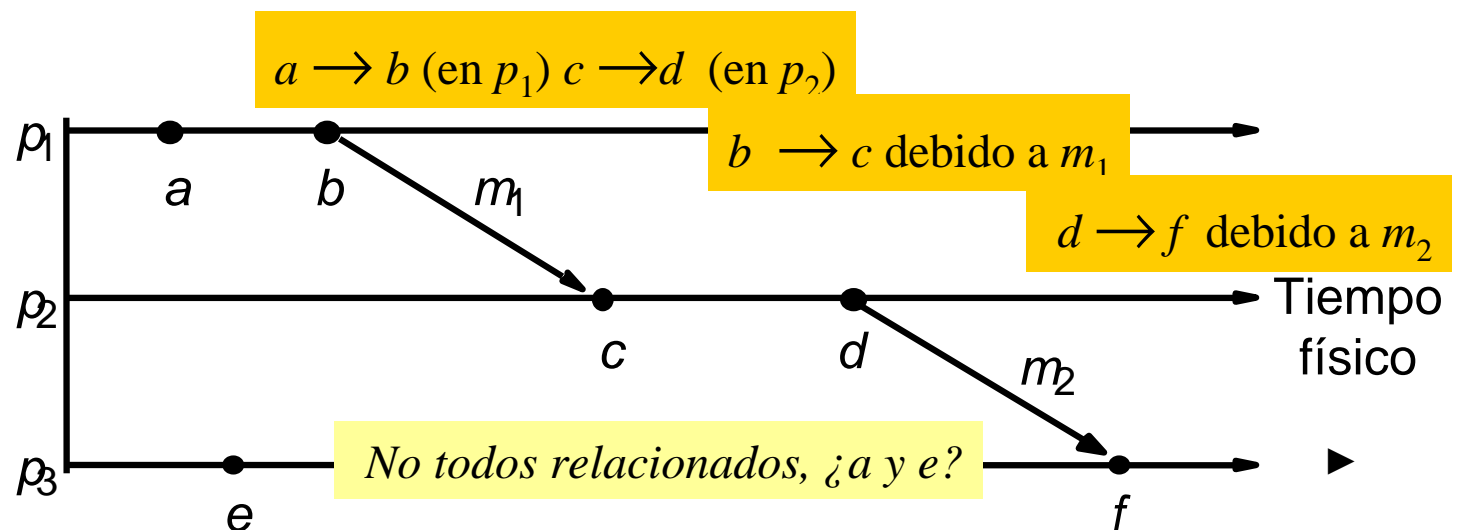
Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Lamport (1978)

Ⓢ No se sincronizan relojes, sino que se ordenan eventos según \rightarrow (relación de orden parcial “suceder antes”):

1. Si los eventos ocurren en p_i ($i = 1, 2, \dots, N$) entonces ocurren en el orden observado por p_i , o sea, \rightarrow_i
2. Cuando m es enviado entre dos procesos, el envío(m) ocurre antes que recepción(m)
3. La relación \rightarrow es transitiva

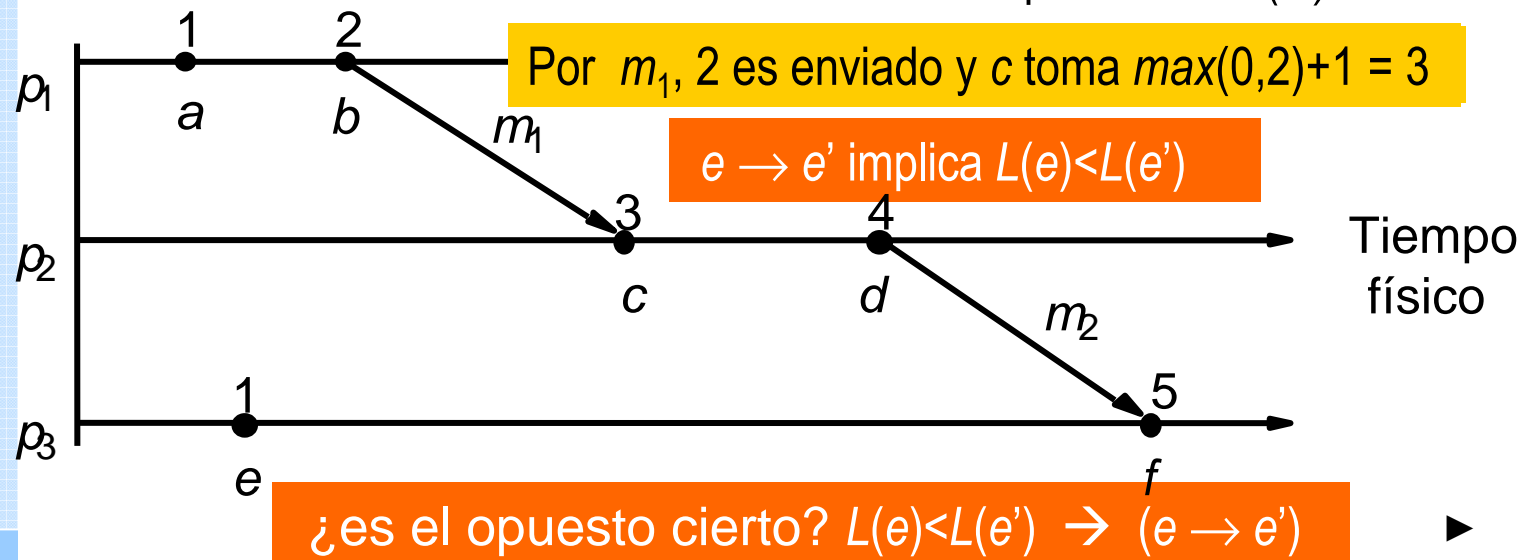


Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Lamport. Relojes lógicos

- Un reloj lógico es un contador software monótono creciente. No se debe confundir con un reloj físico
- Cada proceso p_i tiene su reloj lógico (L_i) que se utiliza para fijar las marcas temporales a los eventos según,
 - R1 L_i se incrementa en 1 antes de cada evento propio de p_i
 - R2.1 cuando p_i envía m , adjunta al mensaje el valor $t = L_i$
 - R2.2 cuando p_j recibe (m, t) establece $L_j := \max(L_j, t)$ y aplica R1 antes de establecer la marca de tiempo de $\text{recibe}(m)$

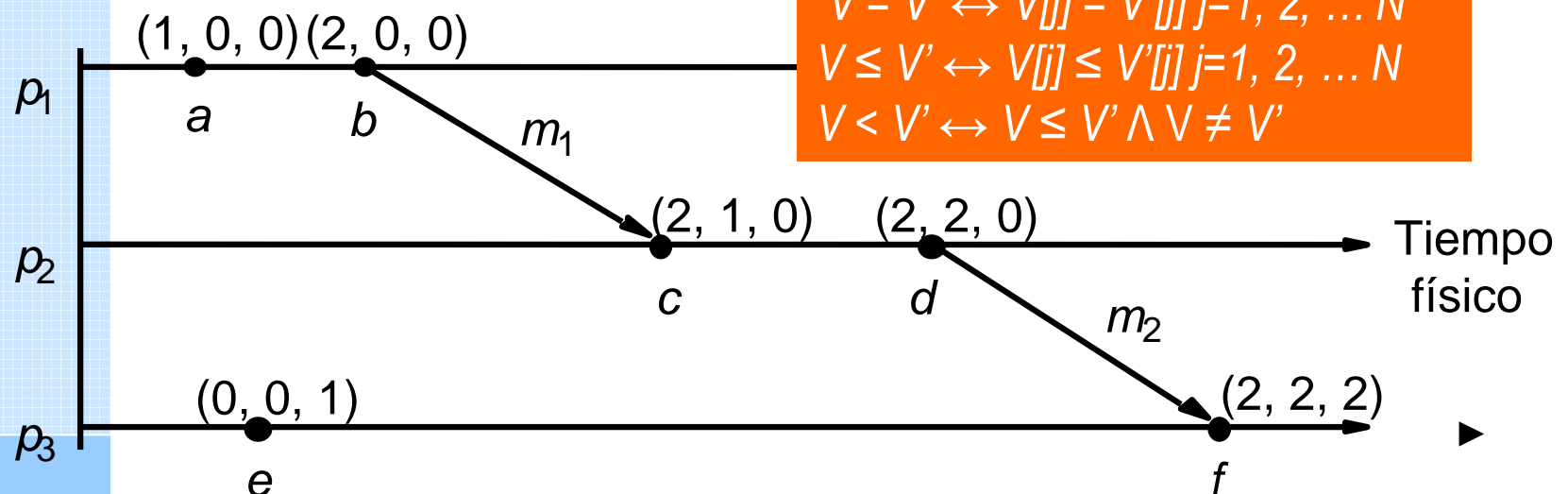


Contenido

introducción
 fundamentos
 tecnologías
 nombres
 tiempo
 seguridad
 coordinación
 transacciones

Relojes vectoriales

- Ⓢ Mattern y Fidge [1989-91] los desarrollan para superar la deficiencia de los relojes lógicos de Lamport: $L(e) < L(e')$ no implica $e \rightarrow e'$
- Ⓢ Un $V_i[i]$ es el número de eventos que p_i ha marcado
 proc $V_i[j]$ ($j \neq i$) número de eventos en p_j que han sido afectados por p_i
 - R1: inicialmente $V_i[j] = 0$ for $i, j = 1, 2, \dots, N$
 - R2: antes de marcar un nuevo evento p_i incrementa $V_i[i] := V_i[i] + 1$
 - R3: p_i adjunta $t = V_i$ en cada mensaje que envía
 - R4: cuando p_i recibe (m, t) establece $V_i[j] := \max(V_i[j], t[j])$ $j = 1, 2, \dots, N$

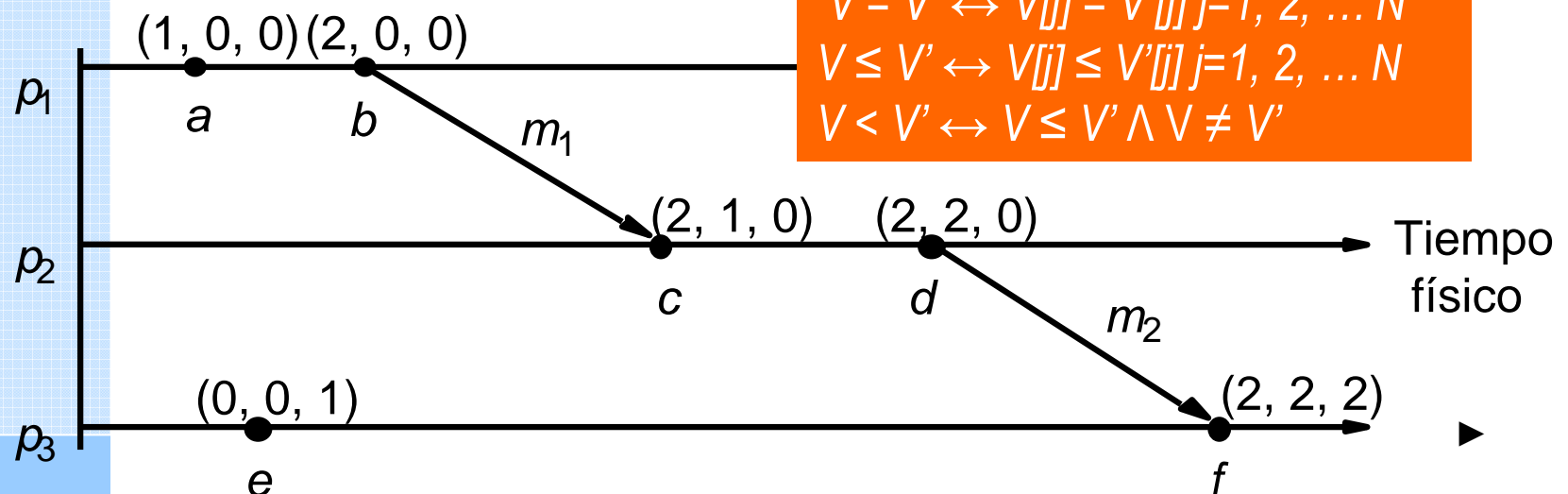


Contenido

introducción
 fundamentos
 tecnologías
 nombres
tiempo
 seguridad
 coordinación
 transacciones

Relojes vectoriales

- Ⓢ Mattern y Fidge [1989-91] los desarrollan para superar la deficiencia de los relojes lógicos de Lamport: $L(e) < L(e')$ no implica $e \rightarrow e'$
- Ⓢ Un reloj vectorial V_i en el proceso p_i es un array de N enteros, que cada proceso utiliza para establecer marcas de sus eventos locales,
 - R1: inicialmente $V_i[j] = 0$ for $i, j = 1, 2, \dots, N$
 - R2: antes de marcar un nuevo evento p_i incrementa $V_i[i] := V_i[i] + 1$
 - R3: p_i adjunta $t = V_i$ en cada mensaje que envía
 - R4: cuando p_i recibe (m, t) establece $V_i[j] := \max(V_i[j], t[j])$ $j = 1, 2, \dots, N$



Contenido

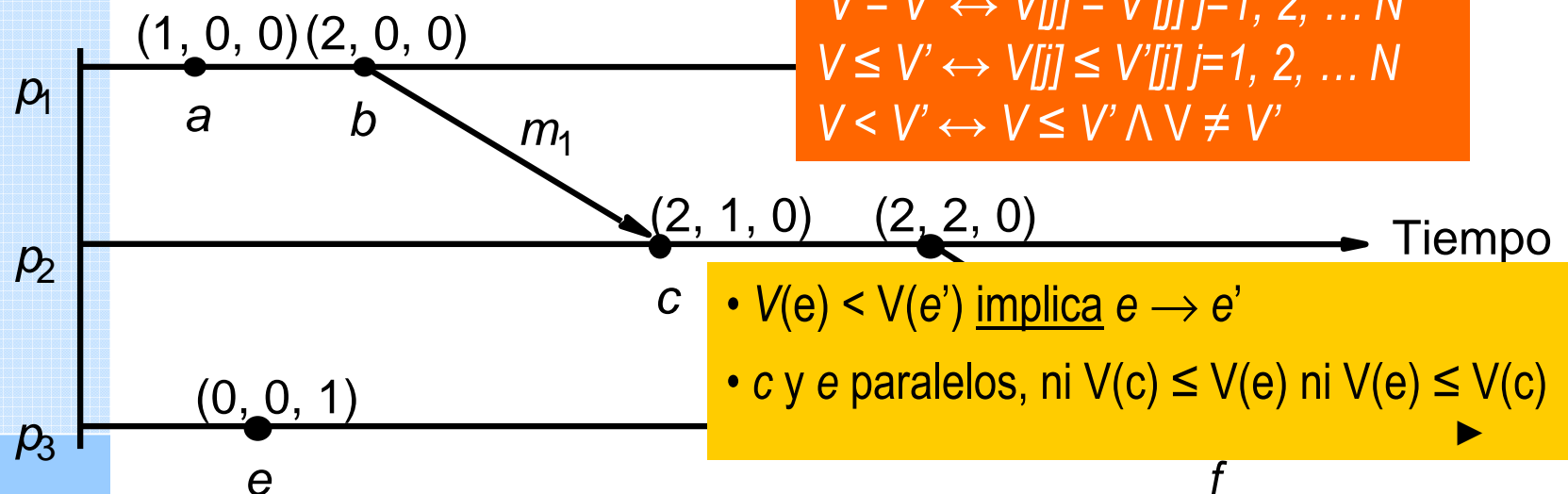
introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Relojes vectoriales

Ⓢ Mattern y Fidge [1989-91] los desarrollan para superar la deficiencia de los relojes lógicos de Lamport: $L(e) < L(e')$ no implica $e \rightarrow e'$

Ⓢ Un r_i $V_i[i]$ es el número de eventos que p_i ha marcado
proc $V_i[j]$ ($j \neq i$) número de eventos en p_j que han sido afectados por p_i

- R1: inicialmente $V_i[j] = 0$ for $i, j = 1, 2, \dots, N$
- R2: antes de marcar un nuevo evento p_i incrementa $V_i[i] := V_i[i] + 1$
- R3: p_i adjunta $t = V_i$ en cada mensaje que envía
- R4: cuando p_i recibe (m, t) establece $V_i[j] := \max(V_i[j], t[j])$ $j = 1, 2, \dots, N$



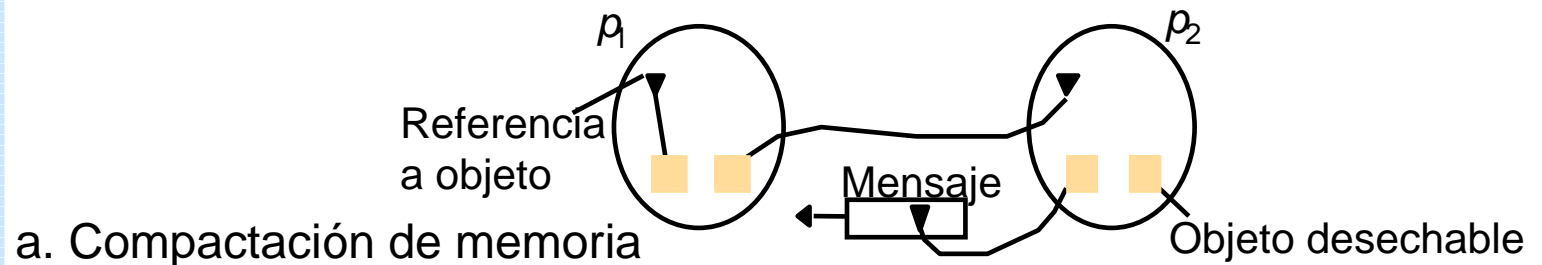
Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

Se examina si una propiedad particular de un SD es cierta cuando éste se ejecuta. Ejemplos:

- Compactación automática de memoria
- Detección distribuida de bloqueos indefinidos
- Detección de terminación distribuida
- Depuración distribuida (no la veremos expresamente)

Ilustran la necesidad de observar el estado del SD globalmente



- Antes de eliminar información se deben analizar la referencias existentes y los canales de comunicación (p.e. mensajes)

Contenido

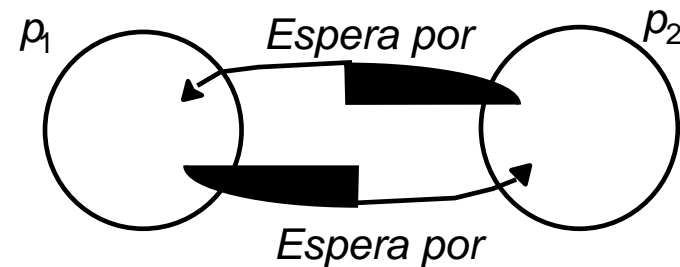
- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

Se examina si una propiedad particular de un SD es cierta cuando éste se ejecuta. Ejemplos:

- Compactación automática de memoria
- Detección distribuida de bloqueos indefinidos
- Detección de terminación distribuida
- Depuración distribuida (no la veremos expresamente)

Ilustran la necesidad de observar el estado del SD globalmente

b. Bloqueo indefinido



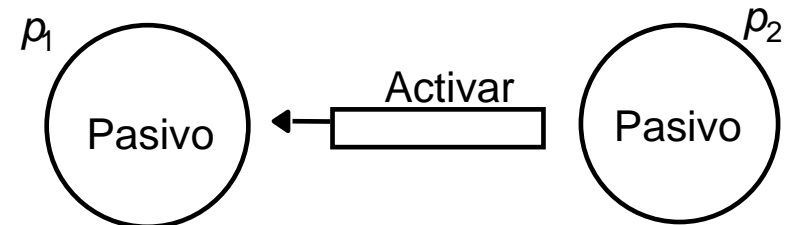
- El “abrazo mortal” clásico, pero entre procesos no ubicados en la misma máquina

Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

- Se examina si una propiedad particular de un SD es cierta cuando éste se ejecuta. Ejemplos:
 - Compactación automática de memoria
 - Detección distribuida de bloqueos indefinidos
 - Detección de terminación distribuida
 - Depuración distribuida (no la veremos expresamente)
- Ilustran la necesidad de observar el estado del SD globalmente

c. Terminación



- Dos procesos pueden en un instante ser pasivos, pero no ser susceptibles de ser eliminados. P.e. P_1 que a pesar de ser pasivo, puede ser activado por un mensaje en tránsito



Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo**
- seguridad
- coordinación
- transacciones

Cortes (defs.)

@ **Historia de un proceso:** $h_i = \langle e_i^0, e_i^1, e_i^2, \dots \rangle$

@ **Prefijo finito** de h_i : $h_i^k = \langle e_i^0, e_i^1, e_i^2, \dots, e_i^k \rangle$

@ **Estado** de p_i antes del suceso k : s_i^k

@ **Historia global del sistema:** $H = h_0 \cup h_1 \cup \dots \cup h_{N-1}$

@ **Corte** de la ejecución del sistema:

$$C = h_0^{c1} \cup h_1^{c2} \cup \dots \cup h_N^{cn}$$

Subconjunto de la historia global que es la unión de los prefijos de las historias de los procesos

@ El estado s_i en el estado global S correspondiente al corte C es el de p_i inmediatamente después del último evento procesado por p_i en C , e_i^{ci} .

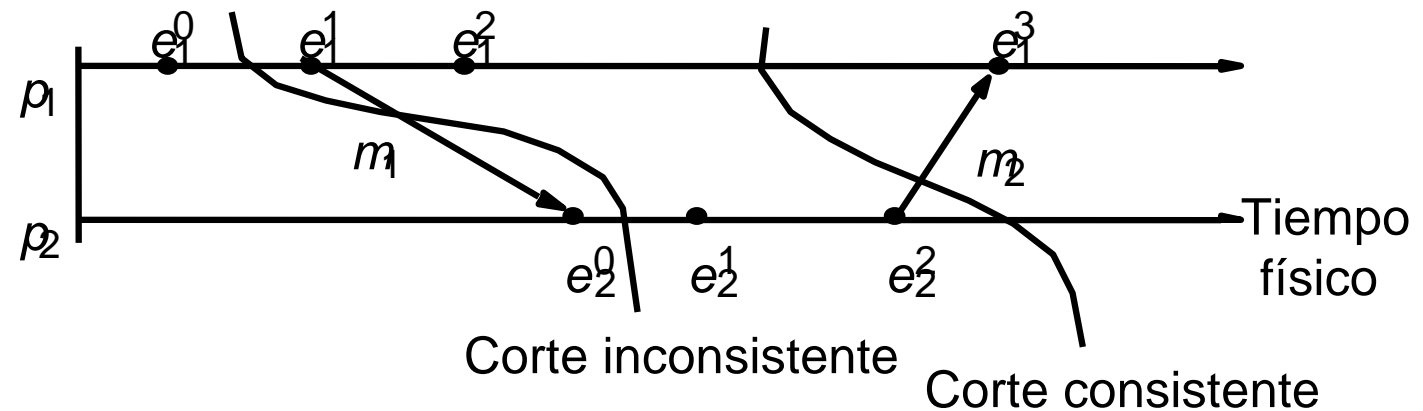
@ **Frontera del corte:** conjunto de sucesos $\{e_i^{ci}\} i = 1, \dots, n$



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

- ④ **Corte consistente:** si, para cada evento que contiene, también contiene todos los sucesos que sucedieron antes de él



- ④ **Estado global consistente:** un estado que corresponde con un corte consistente
- ④ **Caracterización de la ejecución** del SD: transiciones entre estados globales: $S_1 \rightarrow S_2 \rightarrow S_3 \dots$
- ④ **Linealización** o **ejecución consistente:** ordenación de los sucesos en una historia global que es consistente



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Algoritmo de Chandy y Lamport

- ⌚ Algoritmo de instantánea para determinar estados globales de SD
- ⌚ Registra un conjunto de estados de procesos y canales de los procesos del sistema de forma que el estado global sea consistente
- ⌚ Se registra el estado de cada proceso localmente, no hay proceso recolector



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Algoritmo de Chandy y Lamport

Regla de recepción del marcador para el proceso p_i

Cuando p_i recibe un mensaje marcador sobre el canal c :

si (p_i no ha registrado todavía su estado)

- registra su estado de proceso ahora;
- registra el estado de c como el conjunto vacío;
- activa el registro de los mensajes que llegan sobre otros canales entrantes;

sino

- p_i registra el estado de c como el conjunto de mensajes que ha recibido sobre c desde que guardó su estado

fin si

Regla de envío del marcador para el proceso p_i

Después p_i ha registrado su estado para cada canal de salida c :

p_i envía un mensaje marcador sobre c

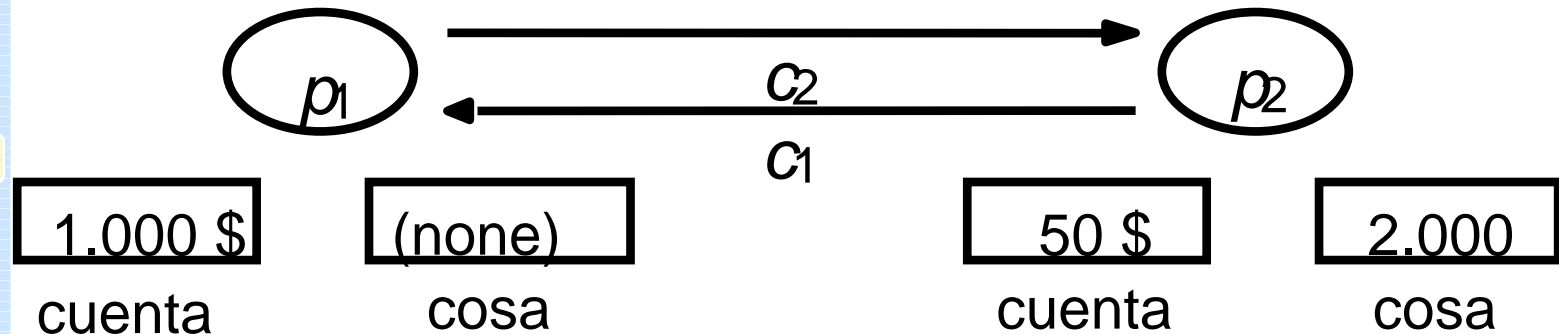
(antes que envíe otro mensaje sobre c).



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Algoritmo de Chandy y Lamport (ejemplo)



- Ⓢ P_1 compra cosas a P_2 , a 10 \$ unidad, a través de c_2
- Ⓢ P_2 sirve cosas a través de c_1

- La figura muestra el estado inicial
- P_2 ya ha recibido una orden por 5 cosas

Contenido

introducción
fundamentos
tecnologías

1. Estado global S_0

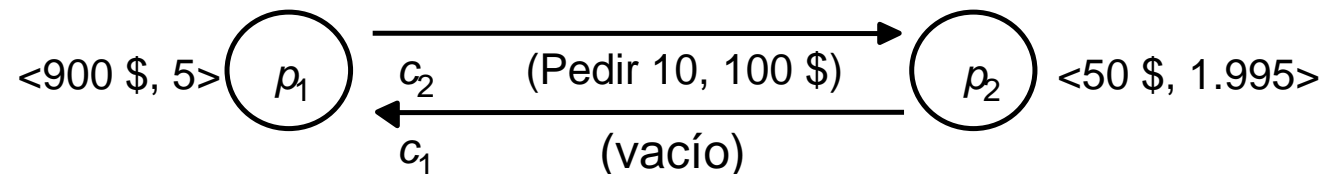
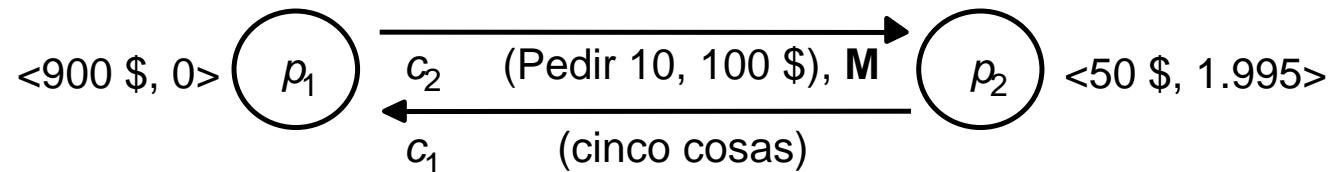
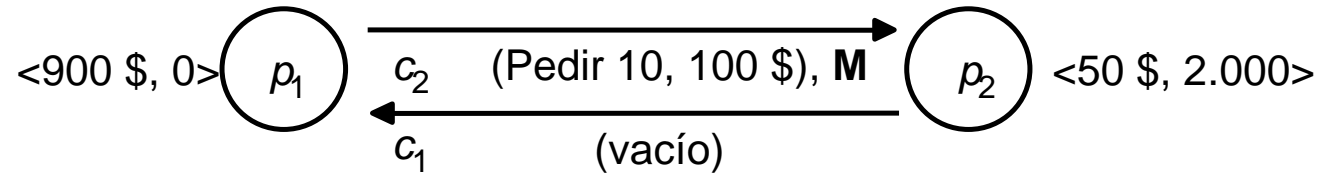
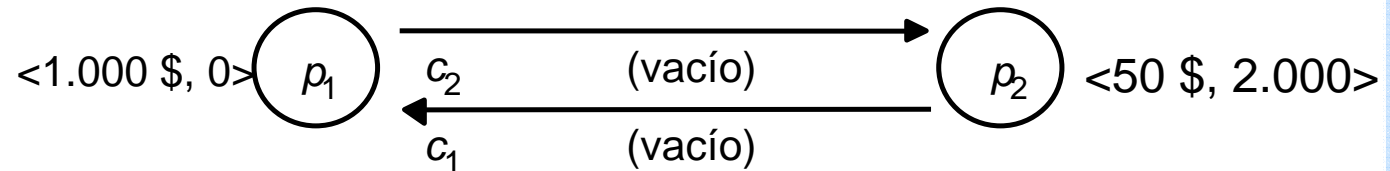
tiempo
seguridad
coordinación

2. Estado global S_1

3. Estado global S_2

4. Estado global S_3

Algoritmo de Chandy y Lamport (ejemplo)



(M = mensaje marcador)

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Algoritmo de Chandy y Lamport

- ⌚ El estado registrado puede diferir de todos los estados globales por los que ha pasado realmente el sistema
- ⌚ El algoritmo selecciona un corte de la historia de la ejecución y cuyo estado registrado es consistente
- ⌚ El algoritmo de instantánea termina, si se cumplen las restricciones de conectividad total e inexistencia de fallo en la comunicación



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

Algoritmo de Chandy y Lamport

(ejercicio propuesto)

- ② Dos procesos P y Q conectados en un anillo utilizando dos canales y permanentemente rotan un mensaje m .
- ② En un instante cualquiera, sólo hay una copia de m en el sistema.
- ② El estado de cada proceso consiste en el número de veces que ha recibido el mensaje m , y P es el que envía el primer mensaje m .
- ② En un cierto punto, P tiene el mensaje y su estado es 101. Inmediatamente después de enviar m , P inicia el algoritmo de instantánea. ►

Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

- ⌚ Se ha puesto de manifiesto la importancia de las marcas temporales en un SD
- ⌚ Los algoritmos como Cristian o NTP sincronizan los relojes a pesar de sus derivas y el retardo de los mensajes
- ⌚ La sincronización de relojes no siempre es suficiente para satisfacer los requisitos de ordenación de dos eventos arbitrarios que sucedan en dos computadores
- ⌚ La relación “suceder antes” es un orden parcial sobre los eventos que reflejan un flujo de información entre ellos
- ⌚ Los relojes de Lamport son contadores que se actualizan de acuerdo con la relación “suceder antes” entre eventos



Contenido

introducción
fundamentos
tecnologías
nombres
tiempo
seguridad
coordinación
transacciones

- ⌚ Los relojes vectoriales mejoran los de Lamport
 - determinan si dos eventos están ordenados por la relación “suceder antes” o son concurrentes, comparando los vectores de marcas
- ⌚ Hemos revisado la idea de estado global de un sistema
- ⌚ Se ha definido corte consistente, linealización, ...
- ⌚ Algoritmo de instantánea de Chandy y Lamport

