
Coordinación y consenso

Sistemas Distribuidos

Objetivos

- Explicar el concepto de elecciones
- Analizar y comprender dos algoritmos de elección.
- Algoritmos de elección
- Explicar el concepto de consenso

Consenso

- Los procesos tienen dificultades para ponerse de acuerdo sobre quién será el nuevo coordinador.
- Protocolos de acuerdo específico
 - Elección: elegir al nuevo coordinador

Elecciones

- Elecciones: **escoger** cuál de los procesos jugará el papel de *servidor* en caso de que éste falle.
- Una variante del algoritmo de *servidor central* para exclusión es que el *servidor* se **elija** entre los procesos existentes.
- Es esencial que todos los procesos estén de **acuerdo** en la elección.
- Si el proceso **servidor de procesos** decide retirarse, entonces se requiere otro procesos de elección del sustituto.

Elecciones

Consideraciones

- Un proceso convoca elecciones mediante **una acción** que indica el inicio del algoritmo de elección
- Un proceso no puede pedir más de una elección por vez.
- El proceso elegido debe ser **único**.

Elecciones

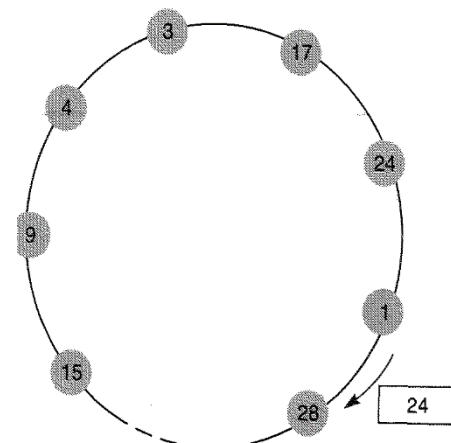
Consideraciones

- Todos procesos tienen un **identificador único**.
- Cada proceso mantiene una variable *elegido* (e) que contendrá el identificador del proceso elegido ($e_i = P$)
- Cuando el proceso se convierta en participante, se fija la variable al valor especial \perp , indicando que no hay consenso todavía. ($e_i = \perp$)

Algoritmos de elección

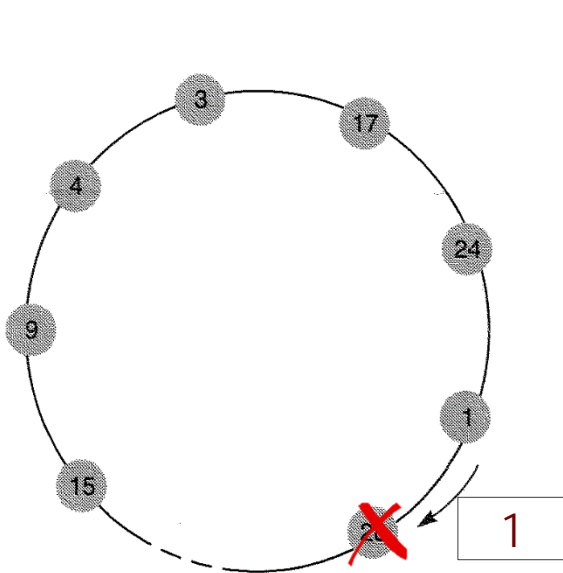
Algoritmo basado en anillo

- Cada proceso tiene un canal de comunicación hacia el siguiente proceso en el anillo.
- El objetivo es elegir un proceso individual llamado *coordinador*, que es el proceso con el **identificador mayor**.
 - Un proceso **convoca a elecciones** y envía un mensaje de «elección» al vecino junto con su identificador.
 - El vecino **compara el identificador** que viene en el mensaje con el suyo: si es mayor cambia el identificador.
 - Cuando llega el mensaje al proceso convocante, **finaliza la elección**, transmitiendo en el mismo orden el identificador elegido.



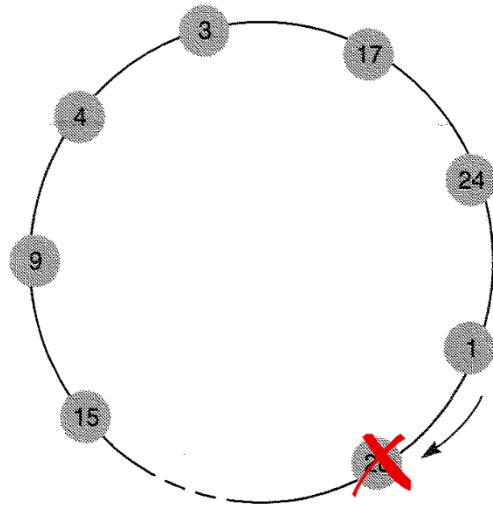
Algoritmos de elección

Algoritmo basado en anillo: ejemplo



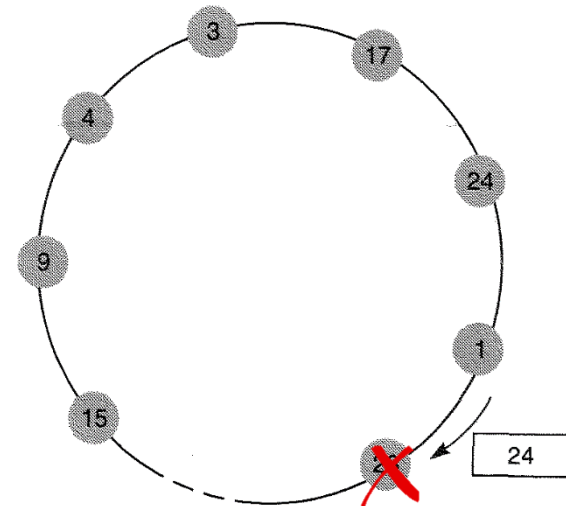
El proceso 1 detecta que el proceso 28 (coordinador) se ha caído y decide convocar una elección.

Envía un mensaje «elección» con su identificador al nodo vecino.



En cada proceso se comparan los identificadores $15 > 1$, cambia el identificador del mensaje y lo envía.

$17 > 15$, $24 > 17$

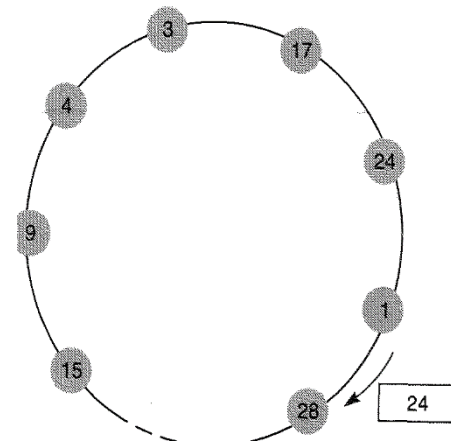


El mensaje llega al proceso que convocó (1), finaliza la elección y manda un mensaje elegido (24) que da una vuelta más al anillo informando que **él es el elegido**.

Algoritmos de elección

Algoritmo basado en anillo: análisis

- Peor caso: el nuevo elegido es el vecino antihorario del convocante
 - $2N-1$ mensajes
 - $N-1$ mensajes elección hasta alcanzar el vecino antihorario
 - N mensajes elegido

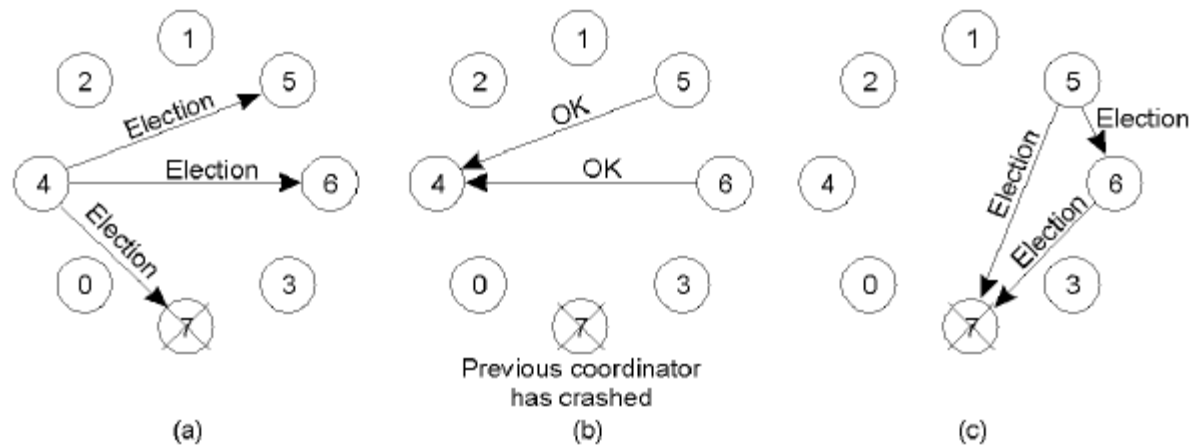


Algoritmos del grandulón

- Utiliza *timeouts* para detectar fallos de procesos
- Asume que cada proceso conoce qué procesos tiene ID mayores y puede comunicarse con ellos.
- Funcionamiento:
 1. Un convocante *inicia una elección* si no recibe respuesta del coordinador en un tiempo T.
 2. El convocante envía mensajes *elección* a los procesos de id mayor
 3. Si ninguno le responde, multidifunde que es el nuevo coordinador
 4. Si alguno le responde, el convocante inicial queda en espera, y los procesos que responden inician un nuevo proceso de elección como convocantes (vuelta al paso 1)

Algoritmos del grandulón

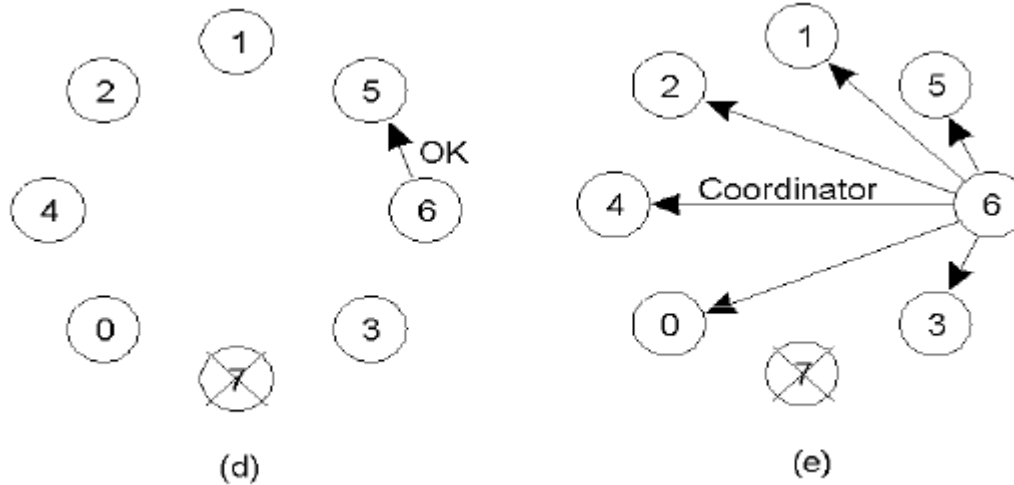
Algoritmo grandulón: ejemplo



- Proceso 7 (coordinador) falla
- Proceso 4 envía elección
- Proceso 5 y 6 responden, diciéndole que pare
- Ahora 5 y 6 comienzan la elección ...

Algoritmos del grandulón

Algoritmo grandulón: ejemplo



d) Proceso 6 dice a 5 que pare

e) Proceso 6 indica a todos que es el coordinador

Ejercicios: Algoritmo del Servidor Central para la Exclusión Mutua

Considere que los siguientes procesos desean utilizar un recurso compartido, bajo el siguiente escenario:

- Inicialmente el recurso está disponible.
- El Proceso 1 solicita el acceso al recurso compartido a las 12:45:07 (hh:mm:ss) y estará usando dicho recurso por 7 segundos.
- El Proceso 2 solicita el acceso al recurso compartido a las 12:45:10 (hh:mm:ss) y estará usando dicho recurso por 5 segundos.
- El Proceso 3 solicita el acceso al recurso compartido a las 12:45:11 (hh:mm:ss) y estará usando dicho recurso por 2 segundos.

Suponiendo que la transmisión de los mensajes no consume tiempo significativo. Realizar la ejecución del algoritmo del servidor central para la exclusión mutua.

- Escribir cada uno de los pasos y el envío de mensajes
- Mostrar el estado de la cola de peticiones en cada paso.
- ¿Cuántos mensajes fueron enviados en total?

Ejercicios

- Crear un ejemplo con, al menos, cuatro procesos para explicar paso a paso, el funcionamiento del algoritmo del grandulón para la elección de un nuevo coordinador dentro de un sistema distribuido.