



Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

COMPUTACIÓN DISTRIBUIDA

Práctica 3

Jorge Angel Sánchez Sánchez

Fecha de entrega: 24 de Mayo 2024

Desarrollo de la práctica:

- definimos dos variables uno que representa la cantidad de nodos y otro la cantidad de tiempo de recuperación de tiempo de los nodos caídos.
- Primero hacemos la estructura de un Nodo con un identificador u un booleano si está activo o no.
- luego hacemos un arreglo de número de nodos
- método imprimirHora: Esta imprime la hora actual en la consola en formato HH:MM:SS.
- método inicializarNodos: inicializa los nodos del sistema asignándoles identificadores únicos y marcándolos como activos. Itera sobre cada nodo en el arreglo.
- método simularFallos: simula la caída de algunos nodos al azar. Utiliza la función rand() de la biblioteca estándar de C para generar un número aleatorio entre 1 y el número total de nodos (NUM NODOS). Luego, marca como inactivos los nodos seleccionados aleatoriamente y muestra sus identificadores en la consola.
- método recuperarNodos: simula la recuperación de los nodos caídos después de un tiempo determinado (RECUPERACION TIEMPO). Utiliza la función sleep(). Luego, marca como activos los nodos que estaban inactivos y muestra un mensaje en la consola indicando que el nodo ha sido recuperado.
- método iniciarElecciones: Comienza por imprimir un mensaje indicando que el nodo inicio ha iniciado el proceso de elección. Luego, itera sobre los nodos siguientes, enviando mensajes de elección a los nodos activos de mayor identificador. Cuando encuentra un nodo activo, invoca recursivamente la función iniciarElecciones() con ese nodo como inicio. Cuando no hay más nodos activos con identificadores mayores, el nodo inicio se declara como el nuevo líder y notifica a los demás nodos activos. Se agregó un booleano respuestaRecibida para controlar si se recibió una respuesta del nodo al que se envió el mensaje de elección. Si no se recibe una respuesta, se considera que el nodo actual es el nuevo líder.
- método main: inicializa los nodos llamando a inicializarNodos() y simula las caídas de nodos llamando a simularFallos(). Luego, busca el primer nodo activo para iniciar el proceso de elección con iniciarElecciones(). Si no hay nodos activos, imprime un mensaje de error. Se agregó una nueva función detectarFalloLider() para determinar si el líder ha fallado y reiniciar el proceso de elección si es necesario. Se llamó a detectarFalloLider() antes de iniciar la recuperación de nodos.
- La simulación de caída y recuperación de nodos se realiza de manera aleatoria al inicio de la ejecución del programa. Las salidas del programa se presentan en la consola y se pueden modificar el número de nodos y el tiempo de recuperación de los nodos caído

Compilación:

- gcc Practica3JorgeSanchez.c -o Practica3JorgeSanchez

Ejecución:

- ./Practica3JorgeSanchez