



FACULTAD DE CIENCIAS
COMPUTACIÓN DISTRIBUIDA

PRACTICA 02

Semestre 2024 – 1

Profesor:

Luis Germán Pérez Hernández

Ayudantes:

Daniel Michel Tavera

Yael Antonio Calzada Martín

Alumnos

Marco Silva Huerta

316205326

Edgar Montiel Ledesma

317317794

Carlos Cortés

420004846

17 de Octubre de 2023

Consenso

Ejecución del programa

Compilar

```
mpicc Practica02_EdgarMontiel_CarlosCortes_MarcoSilva.c -o generales
```

Ejecutar

```
./generales
```

Funcionamiento

1. Definición de constantes y estructuras:

- Se definen constantes, como el número total de generales (`NUM_GENERALES`), el número de generales traidores (`NUM_TRAIDORES`), el número máximo de rondas (`MAX Rondas`), y el número de generales traidores tolerados (`F`).
- Se define una estructura **General** que almacena información sobre cada general, incluyendo su identificación (`id`), si es traidor (`es_traidor`), su voto (`voto`), su mensaje (`mensaje`), y su estrategia de voto (`estrategia`).

2. Función para determinar si la votación es válida (`esVotacionValida`):

- Esta función cuenta los votos a favor de atacar y retirarse y verifica si se alcanza la mayoría requerida para validar la votación.

3. Función para realizar una ronda de comunicación (`realizarRonda`):

- En cada ronda, los generales eligen aleatoriamente si votar por atacar (1) o retirarse (0) según su estrategia.
- El voto se almacena en el campo `voto` y se copia en el campo `mensaje` del general.
- Se ha modificado la función para evitar que los generales voten por sí mismos. Ahora, un general no puede votar por sí mismo.

Si el general actual no es el mismo que el proceso actual, elige un voto aleatorio para el general. Asigna el voto aleatorio al mensaje del general. Si el general actual es el mismo que el proceso actual, establece el voto del general en -1. Asigna el voto aleatorio al mensaje del general. Utiliza la función `MPI Bcast` para comunicar el arreglo generales a todos los procesos.

4. Función para elegir un rey (`elegirRey`):

- Esta función determina cuál de los generales no traidores tiene el ID más alto y lo elige como rey.

5. Función para imprimir el resultado de una ronda (`imprimirResultado`):

- Esta función imprime el número de la ronda actual, muestra información sobre cada general (ID, si es traidor y su voto), y verifica si la votación es válida llamando a la función `esVotacionValida`.

6. Función principal (main):

- Se inicializa el generador de números aleatorios.
- Se crea un arreglo de generales y se establecen sus atributos iniciales, como ID, si son traidores, voto indefinido, mensaje indefinido y estrategia de voto aleatoria.

7. Ciclo principal (while):

- Se realiza un ciclo de rondas de comunicación mientras no se alcance un consenso o se supere el número máximo de rondas definido en `MAX Rondas`.
- En cada ronda, se realiza una ronda de comunicación aleatoria (`realizarRonda`) y se muestra el resultado (`imprimirResultado`).
- Se verifica si la votación es válida. Si es válida, se muestra un mensaje y se rompe el ciclo.
- Si no se alcanza un consenso, se elige un nuevo rey entre los generales no traidores (`elegirRey`).

8. Si se supera el límite de rondas definido en `MAX Rondas`, se muestra un mensaje indicando que se alcanzó el límite de rondas sin consenso.

Pseudocódigo del Algoritmo

Algoritmo del Rey

1. Definir las constantes:
 - `NÚMERO_DE_GENERALES`: número total de generales
 - `NÚMERO_DE_TRAIDORES`: número de generales traidores
 - `F`: número de generales traidores tolerados
2. Crear una estructura General con los siguientes campos:
 - `id` (entero): identificador del general
 - `es_traidor` (booleano): verdadero si el general es traidor, falso si es leal
 - `voto` (entero): 0 para retirada, 1 para ataque
 - `mensaje` (entero): mensaje enviado por el general en la ronda actual
3. Inicializar una lista de generales con `NÚMERO_DE_GENERALES` elementos.
4. Inicializar una variable `REY` con un valor aleatorio en el rango `[0, NÚMERO_DE_GENERALES - 1]`
 - Esto selecciona aleatoriamente a un general como el Rey sin que los demás lo sepan.
5. Para cada general en la lista de generales:
 - Asignar un id único al general.
 - Determinar si el general es traidor (`F` generales serán traidores, incluyendo el Rey).

- Inicializar el voto y el mensaje del general.
6. En cada ronda:
 - Cada general, incluido el Rey, elige su voto (0 para retirada, 1 para ataque) de acuerdo a su estrategia.
 7. Calcular el resultado de la ronda:
 - Inicializar las variables votos_ataque y votos_retirada a 0.
 - Para cada general en la lista de generales:
 - Si el general no es traidor:
 - Incrementar votos_ataque o votos_retirada según el voto del general.
 - Si el general es traidor:
 - Tomar el voto del general según su estrategia.
 8. Verificar si la votación es válida:
 - Calcular la mayoría requerida como $(\text{NÚMERO_DE_GENERALES} / 2) + F$.
 - Si $\text{votos_ataque} \geq \text{mayoría}$ o $\text{votos_retirada} \geq \text{mayoría}$, la votación es válida.
 9. Imprimir el resultado de la ronda y si la votación es válida o no.
 10. Repetir las rondas hasta que se alcance un resultado válido o se llegue a un límite de rondas.
 11. Si se supera el límite de rondas, se considera que no hay consenso y se imprime un mensaje indicando la falta de consenso.
 12. Finalizar el algoritmo.