

## Laborator nr. 11

### Alegerea leader-ului

#### 1 Algoritmul LCR (LeLann, Chang și Roberts)

Algoritmul LCR impune sensul de parcurgere *stânga*→*dreapta* pentru inel. Fiecare unitate de procesare  $p_i$  își va transmite propriul identificator  $id_i$  către vecinul din dreapta, apoi va aștepta un mesaj de la vecinul din stânga. Se disting următoarele 3 cazuri:

1. dacă  $id$ -ul recepționat este mai mare decât  $id$ -ul propriu, atunci se va transmite noul  $id$  (cel recepționat) către dreapta;
2. dacă  $id$ -ul recepționat este mai mic decât  $id$ -ul propriu, atunci procesul curent va ignora acest mesaj (în sensul că nu va transmite noul  $id$  către vecinul din dreapta);
3. dacă  $id$ -ul recepționat este propriul  $id$ , atunci procesul curent se va declara lider și va transmite în inel un mesaj de *terminare*.

Algoritmul 1 expune pseudocodul algoritmului LCR.

- *Notatii:*

- $id_i$  reprezintă identificatorul unității de procesare  $p_i$ .
- $stare_i$  memorează starea curentă a unității de procesare  $p_i$  (*adormit*, *trezit*).
- $statut_i$  memorează starea finală a unității de procesare  $p_i$  (*lider* sau *non\_lider*).

- *Premise:*

- Inițial  $lider = necunoscut$  și  $statut_i = adormit$ .

În Figura 1 este prezentat un exemplu de rulare pentru un inel cu 8 procese.

#### 2 Aplicații

Implementați, utilizând MPI, algoritmul prezentat în laborator. Pentru topologia de tip inel poate fi utilizată o topologie de tip cartezian cu  $o$  linie și  $n$  coloane (unde  $n$  reprezintă numărul de procese pentru care rulați aplicația).

```

1: function LCR_LE( $p_i, stare_i, statut_i, lider$ )
2:   while true do
3:     switch do
4:       case nu primeste nici un mesaj :
5:         if  $stare_i = adormit$  then
6:            $stare_i \leftarrow trezit$ 
7:           trimite  $< "id\_nou", id_i >$  spre dreapta
8:         end if
9:         break
10:      case primeste  $< "id\_nou", id >$  dinspre stanga
11:        if  $id = id_i$  then
12:           $statut_i \leftarrow lider$ 
13:           $lider \leftarrow id_i$ 
14:          trimite  $< "lider", lider >$  catre dreapta
15:          break while
16:        end if
17:        if  $id > id_i$  then
18:          trimite  $< "id\_nou", id >$  catre dreapta
19:        end if
20:        break
21:      case primeste  $< "lider", id >$  dinspre stanga
22:         $statut_i \leftarrow non\_lider$ 
23:         $lider \leftarrow id$ 
24:        trimite  $< "lider", lider >$  catre dreapta
25:        break while
26:    end while
27: end function

```

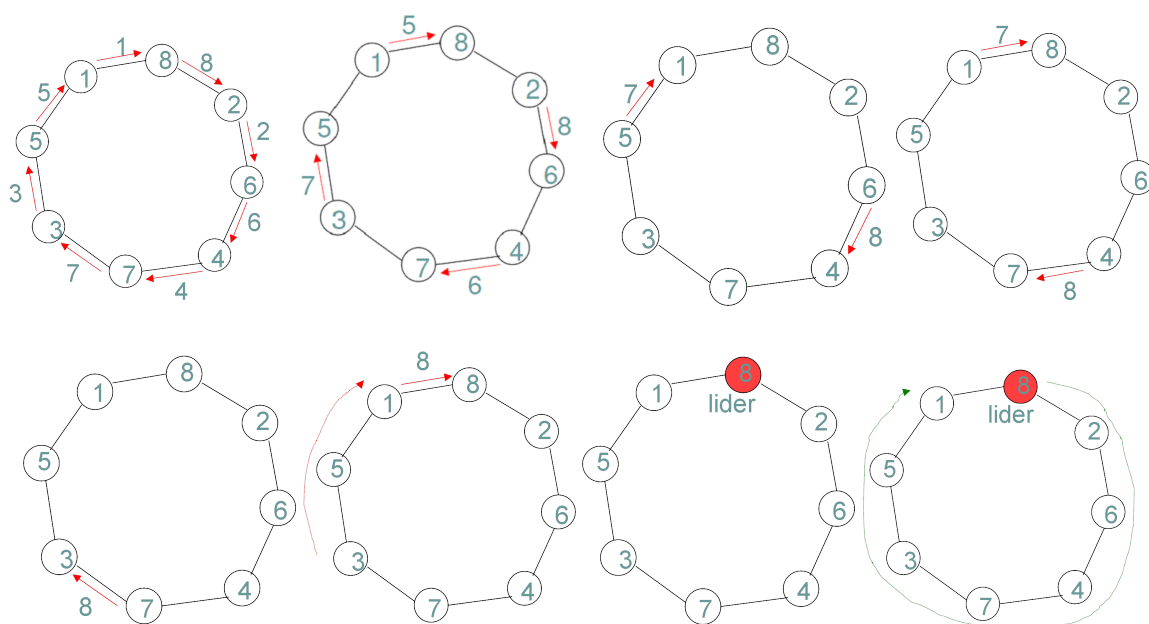


Figura 1: Algoritmul LCR – exemplu de execuție pentru un inel cu 8 procese