

Sistemes Distribuïts – Exercici 2

Exclusió Mútua Distribuïda

Nom(s) de l'estudiant(s)

Curs Acadèmic 2025–2026

Contents

1	Introducció	3
1.1	Objectiu	3
1.2	Enunciat del problema	3
2	Context	4
2.1	Exclusió Mútua Distribuïda	4
2.1.1	Política basada en testimoni	4
2.1.2	Algorisme de Lamport	4
2.1.3	Algorisme de Ricart–Agrawala	4
3	Disseny del sistema	5
3.1	Arquitectura	5
3.2	Esquelets de processos	5
4	Implementació	6
4.1	Fitxers	6
4.2	Procés lleuger Lamport	6
4.3	Procés lleuger Ricart–Agrawala	6
4.4	Controladors pesants	6
5	Flux d’execució	7
5.1	Diagrames de flux	7
5.2	Sortida d’exemple	7
6	Resultats i discussió	8
6.1	Correcció	8
6.2	Reptes de sincronització	8
6.3	Rendiment	8
7	Conclusions	9
7.1	Resum	9
7.2	Treball futur	9

8	Annexos	10
8.1	Codi font	10

Chapter 1

Introducció

1.1 Objectiu

L'objectiu d'aquest exercici és dissenyar i implementar una aplicació distribuïda que demostrï l'exclusió mútua entre processos utilitzant diferents algorismes. Dos processos pesants (HWA i HWB) coordinen processos lleugers (basats en Lamport i Ricart–Agrawala) per accedir de manera segura a un recurs compartit: la pantalla.

1.2 Enunciat del problema

Cada procés pesant crea tres processos lleugers. Tots els processos lleugers s'executen en la mateixa màquina i competeixen per la pantalla. L'exclusió mútua s'aplica de la manera següent:

- Entre processos pesants: política basada en testimoni (token).
- Entre processos lleugers de HWA: algorisme de Lamport.
- Entre processos lleugers de HWB: algorisme de Ricart–Agrawala.

Chapter 2

Context

2.1 Exclusió Mútua Distribuïda

L'exclusió mútua distribuïda garanteix que només un procés entri a la secció crítica alhora en un sistema distribuït, sense control centralitzat.

2.1.1 Política basada en testimoni

Un únic testimoni circula entre els processos pesants. Només el que el posseeix pot permetre que els seus processos lleugers entrin a la secció crítica.

2.1.2 Algorisme de Lamport

Utilitza rellotges lògics i l'ordre dels missatges per decidir l'entrada a la secció crítica. Les sol·licituds s'ordenen per marca temporal i identificador de procés.

2.1.3 Algorisme de Ricart–Agrawala

Cada procés envia missatges REQUEST a tots els seus iguals i espera les respostes REPLY. Els rellotges vectorials determinen la prioritat.

Chapter 3

Disseny del sistema

3.1 Arquitectura

- **Servidor de pantalla:** servidor central que mostra els missatges dels processos.
- **Procés pesant A (HWA):** crea LA1, LA2, LA3 amb l'algorisme de Lamport.
- **Procés pesant B (HWB):** crea LB1, LB2, LB3 amb l'algorisme de Ricart–Agrawala.
- **Comunicació:** tots els processos es comuniquen mitjançant sockets TCP.

3.2 Esquelets de processos

Els processos pesants i lleugers segueixen les directrius de l'exercici:

- **Pesant:** espera el testimoni, activa els lleugers, recull confirmacions DONE, passa el testimoni.
- **Lleuger:** espera el senyal del pesant, sol·licita CS, imprimeix el seu ID deu vegades amb pauses, allibera CS, notifica al pesant.

Chapter 4

Implementació

4.1 Fitxers

- `screen_server.py`: mostra missatges dels processos.
- `lwa.py`: processos lleugers amb Lamport.
- `lwb.py`: processos lleugers amb Ricart–Agrawala.
- `hwa.py`: controlador del procés pesant A.
- `hwb.py`: controlador del procés pesant B.

4.2 Procés lleuger Lamport

Implementat a `lwa.py`. Cada procés manté un rellotge lògic, una cua de sol·licituds i un conjunt de respostes. L'entrada a la secció crítica es concedeix quan la seva sol·licitud és al capdavant de la cua i ha rebut totes les respostes.

4.3 Procés lleuger Ricart–Agrawala

Implementat a `lwb.py`. Cada procés manté un rellotge vectorial. Les sol·licituds es difonen amb marques temporals, i les respostes es difereixen o s'envien immediatament segons la relació de precedència.

4.4 Controladors pesants

`hwa.py` i `hwb.py` creen processos lleugers, gestionen el pas del testimoni i coordinen les confirmacions DONE.

Chapter 5

Flux d'execució

5.1 Diagrames de flux

- Circulació del testimoni entre HWA i HWB.
- Cicle de Lamport (REQUEST/REPLY/RELEASE) entre LA1–LA3.
- Cicle de Ricart–Agrawala (REQUEST/REPLY) entre LB1–LB3.

5.2 Sortida d'exemple

Exemple de missatges a la pantalla:

```
Sóc el procés lleuger [LA1] entrant a la secció crítica.  
Sóc el procés lleuger [LA2] entrant a la secció crítica.  
Sóc el procés lleuger [LB1] entrant a la secció crítica.  
...
```


Chapter 6

Resultats i discussió

6.1 Correcció

La implementació garanteix l'exclusió mútua: només un procés lleuger imprimeix alhora, respectant les regles de Lamport i Ricart–Agrawala.

6.2 Reptes de sincronització

La gestió de sol·licituds concurrents, respostes diferides i l'equitat ha requerit un ús acurat de rellotges lògics i vectorials.

6.3 Rendiment

El sistema mostra la sobrecàrrega de sincronització distribuïda deguda al pas de missatges, però manté la correcció.

Chapter 7

Conclusions

7.1 Resum

Hem implementat una aplicació distribuïda amb dos processos pesants i sis processos lleugers utilitzant els algorismes de Lamport i Ricart–Agrawala. El pas de testimoni assegura la coordinació entre processos pesants.

7.2 Treball futur

Millores possibles:

- Extensió a múltiples màquines.
- Afegir tolerància a fallades.
- Mesurar latència i rendiment.

Chapter 8

Annexos

8.1 Codi font

Tots els fitxers font s'inclouen per separat a la entrega.