

Absence justifié lors du Tp1

PARTIE 1 : PRODUCTEURS-CONSOMMATEURS	1
1.1 Objectif	1
1.2 Implémentation (Classe Buffer)	1
1.3 Analyse des Résultats	1
PARTIE 2 : LE DÎNER DES PHILOSOPHES	2
2.1 Le Problème de l'Interblocage	2
2.2 Solution Implémentée : Rupture de la Symétrie	2
2.3 Résultats	2

PARTIE 1 :

PRODUCTEURS-CONSOMMATEURS

1.1 Objectif

L'objectif est de synchroniser les consommateurs et les producteurs partageant une zone de mémoire commune (tampon circulaire) de taille size. J'ai implémenté le modèle de la "Généralisation 2" présenté en cours qui utilise une exclusion mutuelle fine pour optimiser les accès concurrents.

1.2 Implémentation (Classe Buffer)

Pour respecter les contraintes du cours, j'ai utilisé 4 sémaphores distincts :

1. **Gestion du Flux (Comptage) :**
 - vide (initialisé à size) : Compte les places libres. Bloque le producteur si le tampon est plein.
 - plein (initialisé à 0) : Compte les messages disponibles. Bloque le consommateur si le tampon est vide.
2. **Protection des Indices (Exclusion Mutuelle) :**
 - mutexProd (initialisé à 1) : Protège l'indice d'écriture tête (in).
 - mutexConso (initialisé à 1) : Protège l'indice de lecture queue (out).

1.3 Analyse des Résultats

L'utilisation de deux mutex distincts (mutexProd et mutexConso) permet une meilleure performance

```
---1--- DEMANDE
---2--- DEMANDE
++ AJOUTE ++ 12.645793781045446
++ MIS ++ 12.645793781045446
---1--- A RETIRE 12.645793781045446
++ AJOUTE ++ 17.517119963329883
++ MIS ++ 17.517119963329883
---2--- A RETIRE 17.517119963329883
```

Les traces d'exécution confirment que :

- Le principe FIFO est respecté (gestion correcte de tête et queue).
- Les consommateurs lisent en parallèle sans conflit.
- Aucune donnée n'est perdue ou lue en double.

PARTIE 2 : LE DÎNER DES PHILOSOPHES

2.1 Le Problème de l'Interblocage

Cinq philosophes alternent entre penser et manger. Pour manger, ils doivent acquérir deux fourchettes (ressources partagées).

Dans une approche naïve où chaque philosophe exécute l'algorithme "Prendre Gauche (fg) puis Prendre Droite (fd)", un **interblocage (deadlock)** survient si tous les philosophes prennent leur fourchette gauche simultanément. Le cycle d'attente est fermé et le système se fige.

2.2 Solution Implémentée : Rupture de la Symétrie

Pour prévenir ce blocage, j'ai implémenté une solution asymétrique au niveau de l'initialisation des threads dans le Main :

- Les philosophes 0 à N-2 prennent **Gauche puis Droite**.
- Le dernier philosophe (N-1) prend **Droite puis Gauche**.

Cela brise le cycle d'attente circulaire : le dernier philosophe entrera en compétition pour la fourchette droite (déjà prise ou libre) avant de bloquer la gauche.

2.3 Résultats

Diner des Philosophes: Début

Philosophe 0 pense.

Philosophe 3 pense.

Philosophe 4 pense.

Philosophe 2 pense.

Philosophe 1 pense.

Philosophe 4 a FAIM.

Philosophe 4 a pris GAUCHE (-1).

Philosophe 1 a FAIM.

Philosophe 1 a pris GAUCHE (-1).

Philosophe 4 a pris DROITE (-1).

Philosophe 4 MANGE

Ceci valide l'efficacité de la gestion des sémaphores et l'absence d'interblocage.