



Systemes Concurrents

TP EXCLUSION MUTUELLE

Deuxième Année, Systemes Logiciels

YOUNES SAOUDI

2020-2021

Exclusion Mutuelle

1. **Vérification de l'implémentation**
2. L'ordre des deux lignes 29 et 30 est **important** car si en l'inversant, nous **ne pouvons plus garantir l'exclusion mutuelle** puisque les deux processus pourront s'exécuter **simultanément**
3. Utilisation de la classe `java.util.concurrent.atomic.AtomicBoolean` :
 - a. **Implantation du protocole avec TestAndSet**
 - b. **Version Vivace du protocole** `TestAndSetVivace.java`
 - c. **Comparaison des deux versions par rapport à**
`Synchronized (VersionSynchronized.java)` :

Nombre de Threads	TestAndSet (ms)	TestAndSetVivace (ms)	VersionSynchronized (ms)
3	~40	~50	~40
15	~100	~250	~45
25	~400	~700	~55
40	~600	~1500	~65
50	~1000	~2500	~75

- d. **Comparaison avec Peterson** (`VersionPeterson.java`) :

TestAndSet (ms)	TestAndSetVivace (ms)	VersionSynchronized (ms)	VersionPeterson (ms)
~25	~27	~27	~30

Schéma Producteurs-Consommateurs

Voir Implémentation `ProdConso.java`