R3.05

# Semaphrores

## 1.

Un écrivain ferme un verrou lorsqu’il accède à la mémoire, le bloque quand il écrit et l’ouvre quand il a fini de rédiger.

## 2.

Un lecteur ferme un verrou pour entrer dans la mémoire, il le bloque quand il le lit et le bloque quand il lit.

## 3.

Tant que tkt

Si l’écrivain ouvre un verrou

Alors il le bloque quand il rédige

Tant que l’écrivain n’a pas débloquer le verrou

Alors on le laisse fermer et on refuse les lecteurs

Si le verrou est ouvert

Alors un lecteur pour lire le contenu en fermant le verrou

Tant que le lecteur n’a pas débloqué le verrou

Alors on le laisse fermer et on bloque les autres lecteurs ou écrivains

Si le verrou est fermé on ouvre le verrou

Alors on refait la même chose

Correction :

Programme écrivain :

Tant qu’en marche

Ouvrirmutex(verroubal) //blocage si des lecteurs/écrivains

Ecrire

Fermermutex(verroubal) // debloquage écrivains/lecteurs

Traitement

Fin du tant que

Programme lecteur :

Tant qu’en marche

lockmutex(verroucpt)

Cpt <- cpt+1

Si cpt = 1

lockmutex(verroubal)

unlockmutex(verroucpt)

Lire un message dans la boite

lockmutex(verroucpt)

Cpt<- cpt+1

Si cpt = 0

unlockmutex(verroubal)

unlockmutex(verroucpt)

Ftq

## 4.

# Threads et mutex

SUR VSCODE