Sous MacOS High Sierra

=> l'ordre d'accès à la ressource se fait dans l'ordre d'arrivée, c'est un ordonnancement de type FIFO, équitable pour les lecteurs et écrivains

1- lecteurs favorisés ? non, remarquez que les lecteurs 0, 2 et 1 veulent la ressource pourtant l'écrivain 0 l'obtiendra avant le lecteur 1

2 - écrivains favorisé ? non, remarquez que les écrivains 0 et 1 veulent la

ressource pourtant le lecteur 1 l'obtiendra avant l'écrivain 1

3 - c'est un ordonnancement de type FIFO, remarquez que les lecteurs et

écrivains accèdent à la ressource dans l'ordre où ils l'on demandée (dans

les 5 premières lignes du log)

$ gcc -Wall pthread\_rwlock\_exemple.c

$ ./a.out

Lecteur 0 veut consulter la ressource // 1

Lecteur 2 veut consulter la ressource // 1

Ecrivain 0 veut modifier la ressource // 1 // 2

Lecteur 1 veut consulter la ressource // 1 // 2

Ecrivain 1 veut modifier la ressource // 2

Lecteur 0 consulte la ressource : A // 3

Lecteur 2 consulte la ressource : A // 3

Lecteur 0 a fini de consulter la ressource : A

Lecteur 2 a fini de consulter la ressource : A

Lecteur 2 veut consulter la ressource

Ecrivain 0 modifie la ressource : A // 1 // 3

Lecteur 0 veut consulter la ressource

Ecrivain 0 : maintenant la ressource est B

Lecteur 1 consulte la ressource : B // 2 // 3

Ecrivain 0 veut modifier la ressource

Lecteur 1 a fini de consulter la ressource : B

Ecrivain 1 modifie la ressource : B // 3

Ecrivain 1 : maintenant la ressource est C

Lecteur 1 veut consulter la ressource

Lecteur 2 consulte la ressource : C

Ecrivain 1 veut modifier la ressource

Lecteur 0 consulte la ressource : C

Lecteur 2 a fini de consulter la ressource : C

Lecteur 0 a fini de consulter la ressource : C

Ecrivain 0 modifie la ressource : C

Lecteur 2 veut consulter la ressource

Ecrivain 0 : maintenant la ressource est D

Lecteur 1 consulte la ressource : D

Lecteur 1 a fini de consulter la ressource : D

Ecrivain 1 modifie la ressource : D

Lecteur 0 veut consulter la ressource

Lecteur 1 veut consulter la ressource

Ecrivain 1 : maintenant la ressource est E

Lecteur 2 consulte la ressource : E

Lecteur 0 consulte la ressource : E

Lecteur 1 consulte la ressource : E

Lecteur 2 a fini de consulter la ressource : E

Lecteur 0 a fini de consulter la ressource : E

Lecteur 0 veut consulter la ressource

Lecteur 0 consulte la ressource : E

Lecteur 0 a fini de consulter la ressource : E

Lecteur 1 a fini de consulter la ressource : E

Ecrivain 0 veut modifier la ressource

Lecteur 0 : s'en va

Ecrivain 0 modifie la ressource : E

Ecrivain 0 : maintenant la ressource est F

Ecrivain 1 veut modifier la ressource

Lecteur 2 veut consulter la ressource

Ecrivain 1 modifie la ressource : F

Lecteur 1 veut consulter la ressource

Ecrivain 0 veut modifier la ressource

Ecrivain 1 : maintenant la ressource est G

Lecteur 2 consulte la ressource : G

Lecteur 1 consulte la ressource : G

Lecteur 2 a fini de consulter la ressource : G

Lecteur 1 a fini de consulter la ressource : G

Ecrivain 0 modifie la ressource : G

Ecrivain 0 : maintenant la ressource est H

Ecrivain 0 : s'en va

Ecrivain 1 veut modifier la ressource

Lecteur 2 : s'en va

Ecrivain 1 modifie la ressource : H

Ecrivain 1 : maintenant la ressource est I

Lecteur 1 : s'en va

Consultation et modifications terminées

Sous Ubuntu 14.4

=> les écrivains sont favorisés, de plus, le dernier écrivain à avoir utilize la ressource est prioritaire.

1- lecteurs favorisés ? non, remarquez au début du log comme les écrivains mobilisent la ressource alors que les 3 lecteurs sont en attente

2 - fifo ? non, l'écrivain 1 obtient 2 fois la ressource de suite au lieu de laisser la place à l'écrivain 0 et ce dernier fait pareil au lieu de la laisser au lecteur 2

3 - les écrivains sont favorisés, ils monopolisent la ressource, et remarquez que l'écrivain qui vient de lâcher la ressource la re-vérouille en priorité

$ gcc -Wall pthread\_rwlock\_exemple.c -lpthread

$ ./a.out

Ecrivain 1 veut modifier la ressource // 3

Ecrivain 1 modifie la ressource : A // 2 // 3

Ecrivain 0 veut modifier la ressource

Lecteur 2 veut consulter la ressource

Lecteur 1 veut consulter la ressource

Lecteur 0 veut consulter la ressource

Ecrivain 1 : maintenant la ressource est B // 1

Ecrivain 1 veut modifier la ressource // 1 // 3

Ecrivain 1 modifie la ressource : B // 1 // 2 // 3

Ecrivain 1 : maintenant la ressource est C // 1

Ecrivain 0 modifie la ressource : C // 1 // 2

Ecrivain 1 veut modifier la ressource // 1

Ecrivain 0 : maintenant la ressource est D // 1

Ecrivain 0 veut modifier la ressource // 1 // 3

Ecrivain 0 modifie la ressource : D // 1 // 2 // 3

Ecrivain 0 : maintenant la ressource est E // 1

Ecrivain 1 modifie la ressource : E // 1

Ecrivain 1 : maintenant la ressource est F // 1

Ecrivain 0 veut modifier la ressource // 1

Ecrivain 0 modifie la ressource : F // 1

Ecrivain 1 veut modifier la ressource // 1

Ecrivain 0 : maintenant la ressource est G // 1

Ecrivain 0 veut modifier la ressource

Ecrivain 0 modifie la ressource : G

Ecrivain 0 : maintenant la ressource est H

Ecrivain 1 modifie la ressource : H

Ecrivain 1 : maintenant la ressource est I

Lecteur 2 consulte la ressource : I

Lecteur 2 a fini de consulter la ressource : I

Lecteur 1 consulte la ressource : I

Lecteur 1 a fini de consulter la ressource : I

Lecteur 1 veut consulter la ressource

Lecteur 1 consulte la ressource : I

Lecteur 1 a fini de consulter la ressource : I

Lecteur 1 veut consulter la ressource

Lecteur 1 consulte la ressource : I

Lecteur 1 a fini de consulter la ressource : I

Lecteur 1 veut consulter la ressource

Lecteur 1 consulte la ressource : I

Lecteur 1 a fini de consulter la ressource : I

Lecteur 0 consulte la ressource : I

Ecrivain 0 : s'en va

Lecteur 2 veut consulter la ressource

Lecteur 2 consulte la ressource : I

Lecteur 1 : s'en va

Lecteur 0 a fini de consulter la ressource : I

Ecrivain 1 : s'en va

Lecteur 2 a fini de consulter la ressource : I

Lecteur 0 veut consulter la ressource

Lecteur 0 consulte la ressource : I

Lecteur 0 a fini de consulter la ressource : I

Lecteur 2 veut consulter la ressource

Lecteur 2 consulte la ressource : I

Lecteur 0 veut consulter la ressource

Lecteur 0 consulte la ressource : I

Lecteur 0 a fini de consulter la ressource : I

Lecteur 0 veut consulter la ressource

Lecteur 0 consulte la ressource : I

Lecteur 0 a fini de consulter la ressource : I

Lecteur 2 a fini de consulter la ressource : I

Lecteur 0 : s'en va

Lecteur 2 veut consulter la ressource

Lecteur 2 consulte la ressource : I

Lecteur 2 a fini de consulter la ressource : I

Lecteur 2 : s'en va

Consultation et modifications terminées