

TP N°5 – Inversion de priorité

Temps :

Temps estimé à 2h et fini en 1h40 :

- 20 min de réflexion
 - 20 min pour le chronogramme (sur pc)
 - 30 min pour l'implémentation
- 15 min pour les tests
 - 15 min pour le rapport

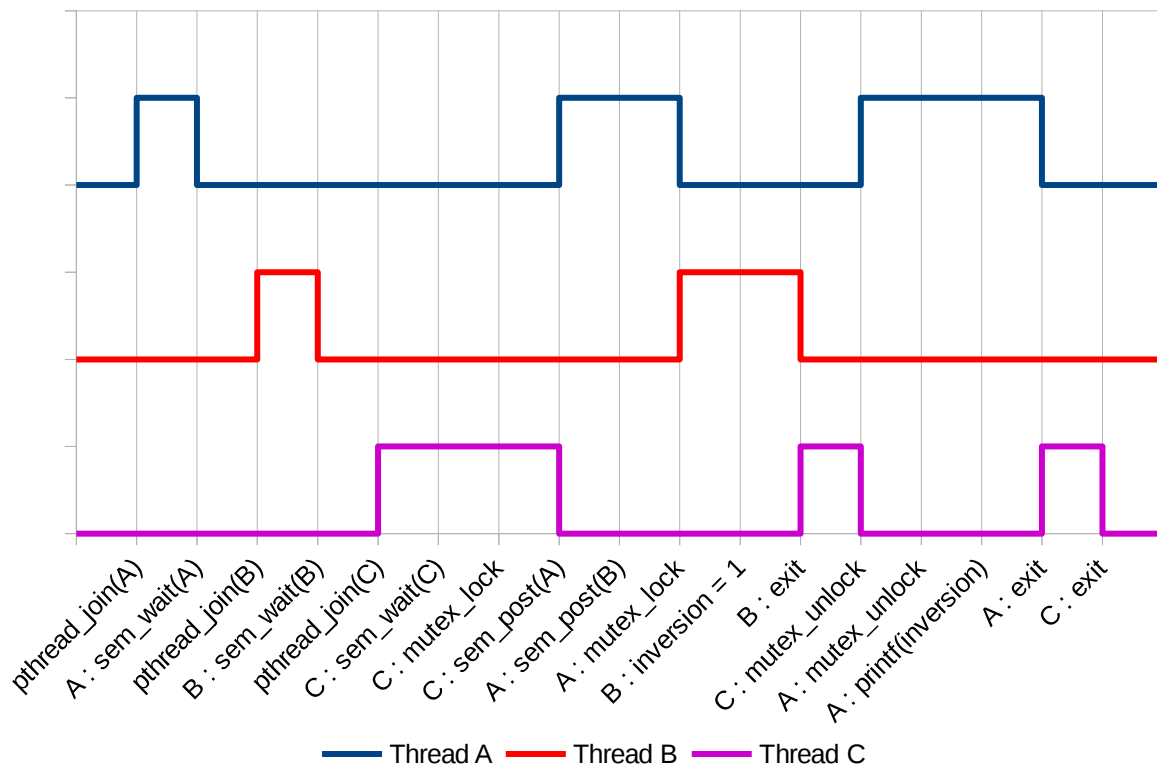
Exécution : \$ make

- Par défaut on effectue une inversion de priorité : `make runHeritage` ou `sudo ./tp5 0`
Pour ne pas avoir d'inversion de priorité, l'argument 1 est nécessaire : `make runNoHeritage` ou `sudo ./tp5 1`
- Pour les tests : `make check` puis `make runCheck` (ceci lancera 1000 fois chaque test)
le nombre de test est personnalisable : `sudo ./check [nb]`
- `make install` pour les checks

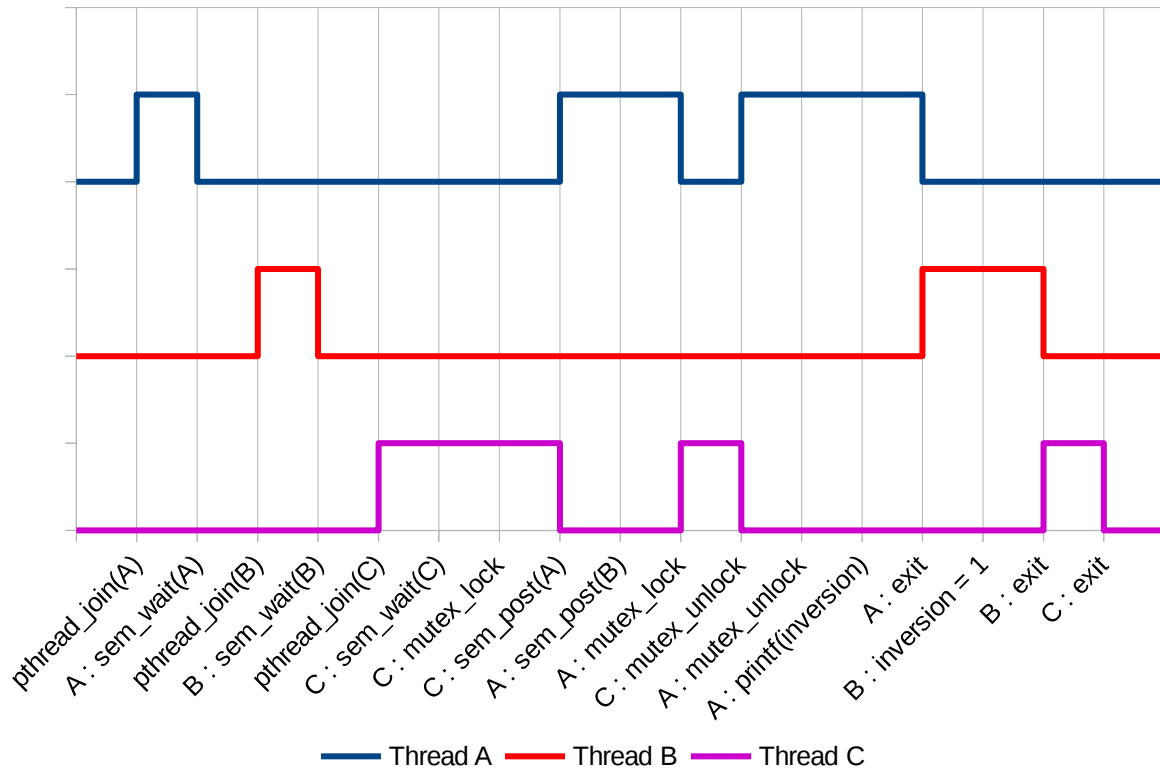
Chronogramme :

Les sémaphores sont initialisés à 0 zéro pour A et B, et à 1 pour C enfin d'être sûr que c'est C qui s'exécute en premier. Les temps de changements de contexte entre les threads sont considérés comme nul sur le chronogramme même si je sais que c'est pas vraiment le cas :).

Chronogramme d'exécution des threads à travers le temps avec héritage



Chronogramme d'exécution des threads à travers le temps sans héritage



Vérification du programme : Dossier Test

Voici l'output que vous devriez obtenir (je l'ai mis au cas où) :

```
$make runCheck
```

```
100 %: Checks : 2, Failures : 0, Errors:0
```

```
Test/check_priority.c : 14 : P : StandardThread : heritage_1 : 0 : Passed
```

```
Test/check_priority.c : 14 : P : StandardThread : heritage_0 : 0 : Passed
```