



Systeme Unix

TP Inter-Processus Communication (IPC)

Contact b-psu-330@epitech.eu

Abstract:

*Ce TP a pour objectif de vous initier à la manipulation des différents IPC afin de vous donner les premiers éléments techniques vous permettant de réaliser votre projet **lemipc**.*



Table des matières

.1	Introduction	2
.2	Etape 1 : Semaphores	3
.3	Etape 2 : Shared Memory	6
.4	Etape 3 : Message queues	6



.1 Introduction

Dans ce TP nous allons voir des exemples d'application des :

1. Semaphores (sem)
2. Shared Memory (shm)
3. Message Queues (msg)

Nous allons pour cela reprendre pour exemple le scénario des trains présenté au cours du TP sur les threads et mutex, en faisant circuler `NB_TRAINS` et en le laissant passer que `NB_PLACES` sur le pont.

Passons donc sans plus attendre à l'étape 1 !



.2 Etape 1 : Semaphores

Reprenez le principe des trains présenté lors du TP sur les threads et écrivez le programme limitant le passage sur un pont à `NB_TRAINS` trains, toujours avec des threads, mais en utilisant une semaphore pour limiter l'accès au pont.



Utilisez une thread pour l'affichage, cela vous sera utile pour le reste du TP.

Déclarez des macros pour les valeurs suivantes :

- `NB_TRAINS` : Le nombre de trains (et donc de voies)
- `NB_PLACES` : Le nombre de trains pouvant passer simultanément sur le pont
- `PONT_DEB` : La position du début du pont
- `PONT_FIN` : La position de la fin du pont
- `SIZE_TOTAL` : La taille totale que le train parcourt

Voici un exemple de valeurs à utiliser :

```
#define NB_TRAINS      7
#define NB_PLACES      3
#define PONT_DEB       3
#define PONT_FIN       5
#define SIZE_TOTAL     9
```

Le résultat attendu devrait ressembler à cela :



```
$> ./sem
semval: 3
0: #==|-|===
1: #==|-|===
2: #==|-|===
3: #==|-|===
4: #==|-|===
5: #==|-|===
6: #==|-|===
semval: 3
0: ==#|-|===
1: ==#|-|===
2: ==#|-|===
3: ==#|-|===
4: ==#|-|===
5: ==#|-|===
6: ==#|-|===
semval: 3
0: ===#-|===
1: ===#-|===
2: ===#-|===
3: ===#-|===
4: ===#-|===
5: ===#-|===
6: ===#-|===
semval: 0
0: ===|#|===
1: ===|#|===
2: ===|#|===
3: ===#-|===
4: ===#-|===
5: ===#-|===
6: ===#-|===
semval: 0
0: ===|-|#==
1: ===|-|#==
2: ===|-|#==
3: ===|#|===
4: ===|#|===
5: ===|#|===
6: ===#-|===
semval: 0
0: ===|-|=##=
1: ===|-|=##=
2: ===|-|=##=
3: ===|-#===
4: ===|-#===
5: ===|-#===
6: ===#-|===
```



```
semval: 2
0: ===| -|===#
1: ===| -|===#
2: ===| -|===#
3: ===| -|#==
4: ===| -|#==
5: ===| -|#==
6: ===|#|===
semval: 2
0: ===| -|===#
1: ===| -|===#
2: ===| -|===#
3: ===| -|#==
4: ===| -|#==
5: ===| -|#==
6: ===| -#===
semval: 3
0: ===| -|===#
1: ===| -|===#
2: ===| -|===#
3: ===| -|===#
4: ===| -|===#
5: ===| -|===#
6: ===| -|#==
semval: 3
0: ===| -|===#
1: ===| -|===#
2: ===| -|===#
3: ===| -|===#
4: ===| -|===#
5: ===| -|===#
6: ===| -|#==
semval: 3
0: ===| -|===#
1: ===| -|===#
2: ===| -|===#
3: ===| -|===#
4: ===| -|===#
5: ===| -|===#
6: ===| -|===#
semval: 3
0: ===| -|===#
1: ===| -|===#
2: ===| -|===#
3: ===| -|===#
4: ===| -|===#
5: ===| -|===#
6: ===| -|===#
$>
```



.3 Etape 2 : Shared Memory

Modifiez le programme précédent pour séparer le traitement en 2 processus. Le premier processus s'occupera du déplacement des trains et le second de l'affichage. Utilisez un segment de mémoire partagée pour stocker la position des trains.



Par la suite, pour complexifier un peu les choses, créez un processus pour l'affichage et autant de processus que de trains.

.4 Etape 3 : Message queues

Reprenez les sources de l'exercice précédent, mais utilisez cette fois les message queues pour informer le processus d'affichage du changement de position des trains.



un message de type 0 est invalide.

Ce TP est maintenant fini. Vous devriez donc être en mesure de commencer dès à présent votre projet `lemipc`.

Merci d'avoir suivi ce TP jusqu'au bout et bon courage pour votre projet.