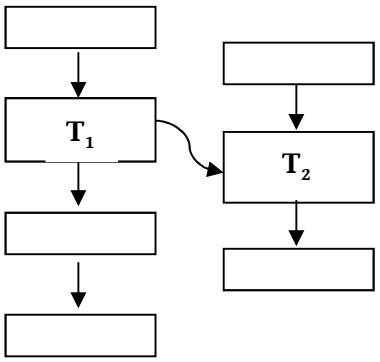
TRAVAUX PRATIQUES SEMAPHORES

EXERCICE 1 - Précédence de tâches

Un système est composé de deux tâches (traitements) T_1 et T_2 soumises à la contrainte de précédence $T_1 < T_2$. Ces deux tâches appartiennent à deux processus différents qui doivent être synchronisés ⇒ **Le deuxième processus doit retarder l'exécution de la tâche T_2 jusqu'à ce que le premier processus termine la tâche T_1.**



EXERCICE 2 – Rendez-vous à 2 et à 3

Deux processus **P1** et **P2** souhaitent établir un rendez-vous avant l'exécution de la fonction f**RendezVous_1()** pour l'un et f**RendezVous_2()** pour l'autre. En utilisant les sémaphores, écrire les programmes **P1.c** et **P2.c** permettant d'établir ce rendez-vous.

Version 2 : Rendez-vous à 2 et à 3 : Réaliser un rendez-vous entre 3 processus **P1**, **P2** et **P3**.

EXERCICE 3 - Le Rendez-vous Emetteurs/Récepteurs

Pour réaliser un mécanisme de communication un à plusieurs, on utilise un ensemble de processus composé d'émetteurs et de récepteurs. Un émetteur produit un message (**à simuler par l'affichage d'un message sur écran**) et se met en attente jusqu'à ce qu'il y ait **n-1** récepteurs au rendez-vous. Un récepteur lancé attend l'émission et l'arrivée des autres récepteurs. Prenons l'exemple d'un rendez-vous **1** à **2** :

1. **$> recepteur &** // Le récepteur 1 se met en attente
2. **$> emmeteur i &** // L'émetteur 2 affiche le message i et se met en attente
3. **$> emmeteur j &** // L'émetteur 3 produit le message j et se met en attente
4. **$> recepteur &** // Le récepteur 4 débloque 1 et 2.
5. **$> recepteur &** // Le récepteur 5 au Rendez-vous avec l'émetteur 3 - Attente
6. **$> recepteur &** // Débloque 3 et 5.