

Présentation

Le but de ce TP est la mise en application des concepts liés à la synchronisation et la communication des processus à travers l'utilisation d'outils offerts par le système d'exploitation LINUX. On se propose d'implémenter un système simplifié d'inscription à l'université utilisant une méthode qui sera décrite par la suite :

- A tous moment, chaque étudiant non inscrit peut envoyer sa demande d'inscription à l'administration. Il peut demander une filière *donnée* ou une filière *appropriée* à sa moyenne fournie.
- L'administration essaye d'affecter l'étudiant selon son choix dans le cas où il exige une filière donnée, si sa moyenne le permis dans la limite du nombre de places disponibles. Dans le cas où l'étudiant n'exige pas de filière, l'administration l'affecte à la filière adéquate à sa moyenne.
- Chaque étudiant peut désister d'une inscription demandée préalablement en envoyant un désistement à l'administration. Suite à deux désistements, l'étudiant (, sous certaines conditions,) peut recevoir une exclusion.
- L'ordre de priorité des filières est définit comme suit :

Soit des constantes $Max_i, i=1, N$ des entiers ≥ 10 tel que : $Max_1 > Max_2 > Max_3 > \dots > Max_N$

Si la moyenne fournie est $\geq Max_1$, la filière appropriée est *Filière₁*,

Si la moyenne fournie est $\geq Max_2$ et $< Max_1$, la filière appropriée est *Filière₂*,

Si la moyenne fournie est $\geq Max_3$ et $< Max_2$ la filière appropriée est *Filière₃*,

Etc...

Si le nombre de places disponibles est nulle pour une filière sélectionnée *i*, le gérant examine la filière *i-1* dans la limite du nombre de places et ainsi de suite.

Si l'étudiant désire une filière *Filière_i*, il sera affecté à cette filière (dans la limite du nombre de places) si sa moyenne est $\geq Max_i$ sinon une réponse négative est retournée à cet étudiant.

Pour tous les cas, si l'étudiant ne peut être affecté par manque de places, le système retourne une réponse *complet*.

Modélisation

Les éléments qui interviennent dans le problème sont :

- Le gestionnaire d'affectation est implémenté par un processus *Affect* cyclique qui reçoit les requêtes des processus dans un tampon nommé *SReq* de taille $N=4$. Ce processus traite les requêtes une à une selon la méthode indiquée ci-dessus en tenant compte de l'état actuel du système. Quatre réponses sont possibles : soit que l'inscription est réalisée et dans ce cas une réponse *positive* avec la filière affectée est retournée au processus ; soit que l'inscription n'est pas réalisée et dans ce cas il envoie une réponse *négative* ou *complet* ; soit que l'étudiant est exclus et dans ce cas il envoie l'indication *exclus*. L'exclusion est décidée si l'étudiant désiste à deux reprises suite à deux affectations à la même filière. La réponse quelle qu'elle soit est envoyée dans une file de messages Fifo nommée *QRep* commune à tous les processus.
- Les étudiants sont implémentés par deux processus cycliques *EtudiantF* et *EtudiantN* qui chacun a pour rôle de simuler le comportement d'un ensemble d'étudiants. Le processus *EtudiantF* représente les étudiants qui exigent une filière donnée et *EtudiantN* représente les étudiants qui n'exigent pas de filière. Chacun de ces deux processus dispose d'un fichier local *FichF* (resp. *FichN*) contenant la suite de requêtes d'un ensemble d'étudiants. Pour chaque requête rencontrée dans le fichier correspondant, le processus *EtudiantF* (resp. *EtudiantN*) formule une requête et l'envoie au processus *Affect* dans le tampon *SReq* et une attente temporaire est réalisée le temps qu'une réponse arrive. Lors de la réception de la réponse, selon la nature de celle-ci, le processus, soit passe à la prochaine requête, soit si la réponse est positive mais ne correspond pas à la filière requise, il peut envoyer un désistement puis reprend cette même requête. Notons aussi qu'un processus peut envoyer un désistement sans condition. Dans le cas de réponse complet, si l'étudiant *EtudiantF* a le choix de demander une autre filière, il formule une nouvelle requête.
- On suppose que le nombre de Filières est de $Nbfilières=3$, le nombre d'étudiants de chaque classe est fixé à $Nbetud=10$, le nombre de places disponibles pour chaque filière est fixé à $Nbplaces=5$.

Travail demandé

Il s'agit de synchroniser tous ces processus à l'aide de sémaphores dans le but de réaliser la fonction décrite ci-dessus. Le système doit comprendre une interface (textuelle seulement, sous forme d'impressions) permettant de suivre et de comprendre le fonctionnement des différentes entités du service objet de ce TP.

Connaissances requises sous Unix

- Création de processus
- Partage de mémoire
- Synchronisation par sémaphores
- Communication par files de messages

Echéance

05 Décembre 2022.