

TP n°4 de Janvier 2017
Synthèse des notions vues en cours, TP1 à 3 et TD 1 à 3

TRES IMPORTANT : Vos comptes-rendus (au format Word97, PDF, ou texte simple) doivent impérativement être envoyés par mail à cdeblang@sfr.fr avec le nom des participants du groupe dans le nom des fichiers fournis en PJ de votre mail.

Le rendu devra être effectué en fin de séance

Ce rendu comprendra le texte source de vos programmes ET la copie d'écran de l'exécution de ces programmes

-

Exercice 1 – Echange de données entre deux tâches à l'aide d'une mémoire partagée et synchronisation par un sémaphore
--

Exposé du problème

- Echanger des données à travers une mémoire partagée, en bloquant toute lecture non précédée d'au moins une écriture à l'aide d'un sémaphore privé. La situation EcritureA/EcritureB/LectureB/LectureA est toutefois possible bien que non satisfaisante.

Travail demandé

Ecrire trois programmes en s'inspirant des programmes mis œuvre dans les précédents TP/TD

A) CREATIONSHM_SEM.c

Créer une mémoire partagée et un sémaphore privé (La valeur initiale du compteur est ZERO) Afficher à l'écran les identifiants de la SHM et du sémaphore

B) ECRIRESHM_SEM.c réalise la séquence suivante
--

Saisir un texte de moins de vingt caractères au clavier Ecrire le texte saisi dans la mémoire partagée V(SémaphorePrivé) CE PROGRAMME A POUR PARAMETRES LES IDENTIFIANTS (mémoire partagée et sémaphore) affichés par CREATIONSHM_SEM
--

C) LIRESHM_SEM.c réalise la séquence suivante :
--

P(SémaphorePrivé) Lire les 20 premiers octets de la mémoire partagée Afficher le texte lu à l'écran CE PROGRAMME A POUR PARAMETRES LES IDENTIFIANTS (mémoire partagée et sémaphore) affichés par CREATIONSHM_SEM

Rappel :

Pour visualiser un IPC vous pouvez utiliser ipcs depuis le shell
Pour détruire un IPC vous pouvez utiliser ipcrm depuis le shell

Exécuter ces programmes :

CREATIONSHM_SEM est exécuté une fois

ECRIRESHM_SEM et LIRESHM-SEM sont exécutés plusieurs fois pour mettre en évidence le respect de la contrainte entre lectures et écritures
En cas de nécessité, avant de réexécuter CREATIONSHM_SEM, il est plus propre de détruire la mémoire partagée et le sémaphore par ipcrm -m ID_SHM et ipcrm -s ID_SEM depuis le shell

Exercice 2 – Echange de données entre deux tâches à l'aide d'un fichier en n'autorisant une lecture qu'après la réception d'un signal émis par l'écrivain

Travail demandé

Ecrire trois programmes en s'inspirant des programmes mis œuvre dans les précédents TP/TD

CREATION_FICHIER.c

ECRITURE_FICHIER.c réalise la séquence suivante
Saisir un texte de moins de vingt caractères au clavier
Ecrire le texte saisi dans le fichier
Emettre le signal

LECTURE_FICHIER.c réalise la séquence suivante
Attendre l'arrivée du signal
Lire le texte écrit dans le fichier

Exécuter ces programmes :

CREATION_FICHIER est exécuté une fois

ECRIRE_FICHIER et LIRE_FICHIER seront exécutés plusieurs fois pour mettre en évidence le respect de la contrainte
«n'autoriser une lecture qu'après la réception d'un signal émis par l'écrivain »
En cas de nécessité, avant de réexécuter CREATIONSHM_SEM, il est plus propre de détruire le fichier

Exercice 3 – Redirections

Vous disposez d'une fonction `void mystere(void)` compilée dans un fichier `mystere.o`, et disponible ici :

<http://www.esiee.fr/~hilairex/IN4R12/1617/mystere.o>

Cette fonction écrit des lignes sur la sortie standard que vous aimeriez pouvoir récupérer. Toutefois, vous n'avez accès qu'au fichier binaire compilé, et pas à son code source.

1. Ecrivez un programme `test_mystere.c`, dont le `main()` se contentera d'appeler `mystere(void)`. Observez le résultat produit.
2. Commentez l'appel à `mystere()` dans `main()`, puis ajoutez une fonction `void pompe_mystere(char *buf)` qui répond à votre problème : elle devra placer dans `buf`, supposé alloué et suffisamment grand, le contenu de ce qu'aura écrit `mystere()` sur la sortie standard, et retourner le nombre d'octets transférés. ATTENTION : votre fonction devra restituer les ressources fichier (tables des fichiers ouverts + fichiers sur disque) dans l'état où elle les a trouvées.
3. Modifiez votre `main()`, pour qu'il écrive « Voilà ce que j'ai lu : », puis appelle `pompe_mystere()` sur un tableau de 5000 caractères maximum, et enfin affiche le tableau lu.