

3 Heures

Documents autorisés :
notes manuscrites et
polycopié de cours

Systeme

Licence Informatique

Exercice 1 Deux processus disposant chacun d'un tableau de taille N veulent pouvoir s'échanger des nombres de ce tableau de manière à ce que, à la fin de leur execution réciproque, P_1 dispose des N plus petites valeurs et P_2 des N plus grandes.

Pour pouvoir échanger ces nombres, on dispose d'une ressource critique contenant trois cases et d'un processus gestionnaire P_0 . Le fonctionnement général est le suivant. Le processus P_1 (resp. P_2) cherche le maximum (resp. le minimum) dans son tableau et l'écrit dans la première (resp. deuxième) case. Le processus P_0 vérifie que la valeur de la première case est supérieure à la valeur de la seconde. Si c'est le cas, P_0 effectue l'échange des valeurs, sinon il bascule le drapeau de la troisième case à -1 . Chaque processus (P_1 et P_2) récupère maintenant sa nouvelle valeur dans la case respective, l'écrit à la place de l'ancienne, puis recommence le même travail. Si le drapeau de la troisième case est à -1 , les processus n'effectuent aucun échange. Ils écrivent leur tableau et s'arrêtent.

Vous disposez des fonctions suivantes :

```
mini(Tableau d'entiers) --> place
maxi(Tableau d'entiers) --> place
echange(adresse,adresse) --> rien
```

En utilisant les sémaphores n-aires $P_n(S)$, $V_n(S)$, écrivez les algorithmes de P_0 , P_1 et P_2 , de manière à ce que l'exclusion mutuelle pour chaque case soit respectée. Écrire les initialisations nécessaires pour les variables partagées et les sémaphores. Les trois processus doivent se terminer lorsqu'il n'y a plus de données à échanger.

Exercice 2 Expliquez le mécanisme de changement de mot de passe sous Unix. Quelle <<astuce Unix>> est utilisée. Donner un autre exemple classique (en argumentant) de l'utilisation de cette astuce.

Exercice 3 Deux processus désirent s'échanger 100 signaux à tour de rôle. Le fonctionnement peut être le suivant. P_0 est le premier processus à démarrer. Il se met en attente d'un signal (SIGUSR1). Lorsque P_1 démarre, il émet un signal (SIGUSR1) vers P_0 puis se met en attente. P_0 , une fois réveillé par l'arrivée du signal, effectue le même travail.

- Donner un mécanisme d'échange des pid des deux processus afin que ceux-ci puissent s'envoyer des signaux.
 - En prenant bien garde d'éviter les risques d'interblocage, donnez les algorithmes des processus P_1 et P_2 en utilisant les signaux POSIX.
 - Expliciter ces risques (utilisation de pause) en donnant un exemple.
-

Exercice 4 On désire implanter un système Client/Serveur d'échange de fichiers entre processus. Le schéma général est le suivant :

On utilisera uniquement trois files de messages. Une entre les clients et le serveur principal, une entre le serveur et les sous-serveurs et une dernière entre clients et sous-serveurs.

Le principe général est celui-ci :

- Un client envoie au serveur principal trois informations : son identifiant, s'il veut émettre ou recevoir, le fichier à émettre ou recevoir.
- Le serveur principal lit ces informations, se duplique (création d'un sous-serveur) et envoie les informations précédentes dans une file à destination du sous-serveur.
- Le sous-serveur gère la communication avec le client. Ils s'échangent des fichiers en lisant ceux-ci bloc par bloc. On supposera que la taille du bloc est prédéfinie.
- Expliquez comment vous allez faire pour n'utiliser que trois files de messages.
- Peut-on se limiter de manière simple à deux files de messages. Expliquez pourquoi.
- Décrire les algorithmes client et serveur (sous-serveurs)
- On désire maintenant implanter un système de compte-rendu géré par le serveur principal. Celui-ci récupèrera des informations comme la taille du fichier par l'intermédiaire des sous-serveurs. Expliquez comment effectuer ces rapports en utilisant une des files de messages déjà existante. Quelle file doit-on utiliser ? Pourquoi ?

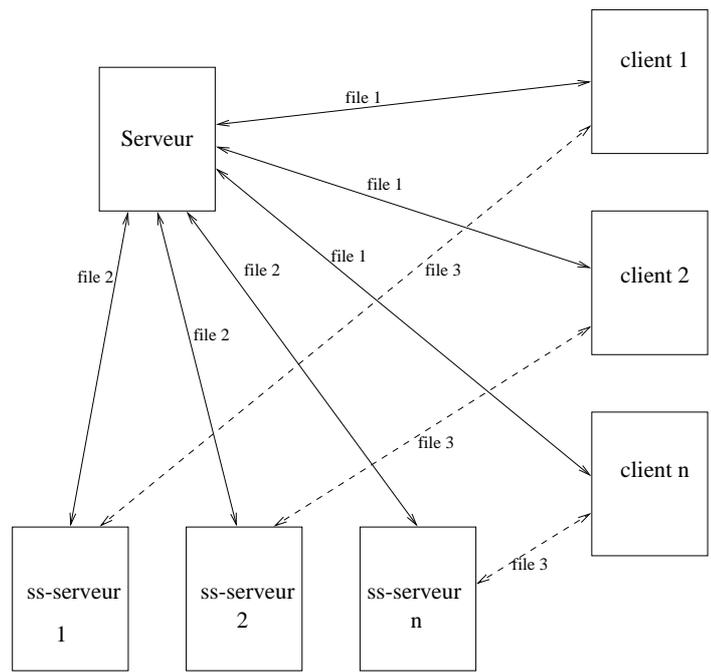


FIG. 1 – Echange de fichiers inter-processus via des files de messages