

Outils IPC

Tableaux de sémaphores

Cet article présente des programmes permettant de manipuler les tableaux de sémaphores IPC System V. Le but est de montrer comment les utiliser pour réaliser un ordre d'exécution entre différents blocs de codes de deux programmes différents.

1 Le serveur

Le serveur exécute trois blocs de code notés A, B et C et le client D, E et F. Le serveur doit exécuter le bloc A avant le bloc E et le client doit exécuter le bloc E avant le bloc C. Le serveur crée d'abord le tableau de 2 sémaphores à l'aide de l'appel système `semget`.

```
int semid;

if((semid = semget((key_t)CLE, 2, S_IRUSR | S_IWUSR | IPC_CREAT |
    IPC_EXCL)) == -1) {
    if(errno == EEXIST)
        fprintf(stderr, "Tableau_de_sémaphores_(cle=%d)_existant\n", CLE);
    else
        perror("Erreur_lors_de_la_création_du_tableau_de_sémaphores_");
    exit(EXIT_FAILURE);
}
```

Il l'initialise ensuite : les valeurs des sémaphores sont fixées à 0.

```
unsigned short val[2] = {0, 0};
if(semctl(semid, 0, SETALL, val) == -1) {
    perror("Erreur_lors_de_l'initialisation_des_sémaphores_");
    exit(EXIT_FAILURE);
}
```

Il exécute le bloc A puis réalise l'opération `V(S0)`. Cela consiste à ajouter 1 au sémaphore.

```
op.sem_num = 0;
op.sem_op = 1;
op.sem_flg = 0;
if(semop(semid, &op, 1) == -1) {
    perror("Erreur_lors_de_l'opération_sur_le_sémaphore_");
    exit(EXIT_FAILURE);
}
```

Il exécute le bloc B. Puis, avant le bloc C, il réalise l'opération $P(S1)$. Cela consiste à soustraire 1 au sémaphore.

```
op.sem_num = 1;
op.sem_op = -1;
op.sem_flg = 0;
if(semop(semid, &op, 1) == -1) {
    perror("Erreur_lors_de_l'opération_sur_le_sémaphore_");
    exit(EXIT_FAILURE);
}
```

2 Le client

Le client récupère le tableau de sémaphores créé par le serveur.

```
if((semid = semget((key_t)CLE, 0, 0)) == -1) {
    perror("Erreur_lors_de_la_récupération_du_tableau_de_sémaphores_");
    exit(EXIT_FAILURE);
}
```

Il exécute ensuite les deux opérations $P(S0)$ et $V(S1)$.

3 Suppression du tableau de sémaphores

Pour supprimer le tableau de sémaphores à l'aide d'un programme en C, nous devons dans un premier temps récupérer l'identifiant interne du tableau à l'aide de `semget`. Puis nous utilisons l'appel système `semctl` avec la commande `IPC_RMID` (le troisième paramètre est inutile).

```
if(semctl(msqid, IPC_RMID, 0) == -1) {
    perror("Erreur_lors_de_la_suppression_du_tableau_de_sémaphores_");
    exit(EXIT_FAILURE);
}
```

4 Compilation et exécution

Le `makefile` fourni permet de compiler les programmes précédents. Dans la section `ARGUMENTS ET COMPILATEUR`, la variable `CLE_SEM` correspond à la clé du tableau de sémaphores utilisé dans les différents programmes. Elle est spécifiée par `gcc` grâce à l'option `-D`.

Pour compiler les programmes précédents, saisissez la commande suivante :

```
make
```

Pour tester les programmes, dans un premier terminal, exécutez le serveur :

```
./semaphoreServeur
```

Dans un second terminal (ou dans un nouvel onglet), exécutez le client :

```
./semaphoreClient
```

Pour supprimer le tableau, vous pouvez soit utiliser le programme `semaphoreSupprime`, soit la règle du `makefile` `cleanIPC`, soit utiliser la commande `ipcrm` (où `X` est la clé du tableau de sémaphores) :

```
./semaphoreSupprime  
make cleanIPC  
ipcrm -Q X
```