

VII : Implémentation des Sémaphores

1. Création de sémaphore : SEMGET

L'appel système **semget()** permet la création d'un sémaphore ou un tableau de sémaphores.

int semget (key_t key, int nsems, int semflg)

nsems : nombre de sémaphore associé à **key**

semflg: option de création (**IPC_CREAT** , **IPC_EXCL**, droits d'accès)

L'appel système **semget()** retourne l'identifiant de l'ensemble de sémaphores associé à la valeur de clé **key**.

Un nouvel ensemble contenant **nsems** sémaphores est créé si **key** a la valeur **IPC_PRIVATE** ou si aucun ensemble n'est associé à **key**, et si l'option **IPC_CREAT** est présente dans **semflg**.
-1 en cas d'erreur

Exemple : `int S = semget(cle,1,0777|IPC_CREAT)`

2. Contrôle de Sémaphore : SEMCTL

int semctl (int semid, int semno, int cmd, union semun)

Effectue l'opération de contrôle indiquée par **cmd** sur le jeu de sémaphores (ou sur le **semno**-ième sémaphore du jeu) identifié par **semid**.

```
union semun { int val;                /* Valeur pour SETVAL */
               struct semid_ds *buf;   /* Tampon pour IPC_STAT, IPC_SET */
               short *array;           /* Tableau pour GETALL, SETALL */
```

semid : identificateur de sémaphore

semno: numéro du sémaphore (Les sémaphores sont numérotés à partir de zéro.)

cmd : commande à exécuter

SETVAL : Initialiser la valeur d'un sémaphore

GETVAL : récupérer la valeur d'un sémaphore

IPC_STAT : Copier les informations de la structure de données noyau associé à semid dans la structure semid_ds

PC_RMID : Supprimer immédiatement le jeu de sémaphores

Exemple semctl() :

```
union semun ctrl;  
ctrl.val = valeur;  
semctl(id,0,SETVAL,ctrl)
```

3. Opérations sur les sémaphores : SEMOP

int semop(int semid, struct sembuf * op, unsigned nsop)

semop() effectue des opérations sur les membres des sémaphores identifié par *semid*.

struct sembuf {

 unsigned short sem_num; /* Numéro du sémaphore */

 short sem_op; /* Opération sur le sémaphore +1 pour P, -1 pou V*/

 short sem_flg; /* Options pour l'opération */

}

 sem_flg : IPC_NOWAIT et SEM_UNDO

Exemple : struct sembuf op;

 op.sem_num = 0;

 op.sem_op = 1;

 op.sem_flg = SEM_UNDO;

 semop(id,&op,1);

4. Module de Gestion des sémaphores

Le module sur les sémaphores (semaphore.h, semaphore.c) permet la gestion des sémaphores.

Il implémente les fonctions permettant :

- La création d'un sémaphore
- L'initialisation d'un sémaphore
- L'opération de blocage d'un processus
- L'opération de déblocage d'un processus

Semaphore.h

```
#include <sys/ipc.h>  
#include <sys/sem.h>  
  
union semun{  
    int val;  
    struct semid_ds *buf;  
    ushort *array;  
};  
  
int Creat_Sem(int key);  
void Init_Sem(int id, int valeurInit);  
void P(int id);  
void V(int id);
```

Semaphore.c

```
#include "semaphore.h"
int Creat_Sem(int key){
    int idsem = semget(cle,1,0777|IPC_CREAT);
    if (idsem == -1) { perror("semget");
                    exit(0);
    }
    return idsem;
}

void Init_Sem(int id, int valeur) {
    union semun ctrl;
    ctrl.val = valeur;
    semctl(id,0,SETVAL,ctrl); }

void P(int id) {
    struct sembuf op;
    op.sem_num = 0;
    op.sem_op = -1;
    op.sem_flg = SEM_UNDO;
    semop(id,&op,1);
}

void V(int id) {
    struct sembuf op;
    op.sem_num = 0;
    op.sem_op = 1;
    op.sem_flg = SEM_UNDO;
    semop(id,&op,1);
}
```