IUP MIAGE
Système d'exploitation

Utilisation des sémaphores

Philippe MARQUET

1 Brefs rappels — Définitions

Un sémaphore est une structure de données composée
- d'un compteur ;
- d'une liste de processus en attente.

Le compteur peut prendre des valeurs entières positives, négatives, ou nulles. Lors de la création d'un sémaphore, le compteur est initialisé à une valeur donnée positive ou nulle ; la file d'attente est vide.

Un sémaphore est manipulé par les deux actions *atomiques* suivantes :

`wait (s)` (aussi nommée `P (s)` ou `down (s)`) décrémente le compteur associé au sémaphore. Si sa valeur est négative, le processus appelant se bloque dans la file d'attente.

`signal (s)` (aussi nommée `V (s)`, `up (s)`, ou `post (s)`) incrémente le compteur. Si le compteur est négatif ou nul, un processus est choisi dans la file d'attente est devient actif.

Deux utilisations sont faites des sémaphores :
- la protection d'une ressource partagée (par exemple l'accès à une variable, une structure de donnée, une imprimante...). On parle de sémaphore d'exclusion mutuelle ;
- la synchronisation de processus (un processus doit en attendre un autre pour continuer ou commencer son exécution).

Bien souvent on peut assimiler la valeur positive du compteur au nombre de processus pouvant acquérir librement la ressource ; et assimiler la valeur négative du compteur au nombre de processus bloqués en attente d'utilisation de la ressource.

2 Exemple : le producteur consommateur

Une solution du problème du producteur consommateur au moyen de sémaphores est donnée à la figure 1. Les deux utilisations types des sémaphores sont illustrées.

3 Exercices

Exercice 1 (Examen de septembre 1995)
Soient 3 processus qui exécutent les programmes suivants :

```
P1: loop A1 end_loop
P2: loop A2 end_loop
P3: loop A3 end_loop
```

Utiliser des sémaphores pour synchroniser ces 3 processus de telle manière que:

Question 1.1 Les actions `Ai` ne soient jamais simultanées.

Question 1.2 Les actions `Ai` ne soient jamais simultanées et se déroulent toujours dans l'ordre `A1A2A3A1A2A3`...

Id: sema.tex,v 1.7 2001/02/15 15:25:30 marquet Exp

```
#define N 100                          /* nombre de places dans le tampon */

semaphore_t mutex = sema_init (1) ;    /* contrôle d'accès au tampon */
semaphore_t vide  = sema_init (N) ;    /* nb de places libres */
semaphore_t plein = sema_init (0) ;    /* nb de places occupées */

void producteur (void)
{
  objet_t objet ;

  while (TRUE) {
    produire_objet (&objet) ;          /* produire l'objet suivant */
    down (&vide) ;                     /* déc. nb places libres */
    down (&mutex) ;                    /* entrée en section critique */
    mettre_objet (objet) ;             /* mettre l'objet dans le tampon */
    up (&mutex) ;                      /* sortie de section critique */
    up (&plein) ;                      /* inc. nb place occupées */
  }
}

void consommateur (void)
{
  objet_t objet ;

  while (TRUE) {
    down (&plein) ;                    /* déc. nb emplacements occupés */
    down (&mutex) ;                    /* entrée section critique */
    retirer_objet (&objet) ;           /* retire un objet du tampon */
    up (&mutex) ;                      /* sortie de la section critique */
    up (&vide) ;                       /* inc. nb emplacements libres */
    utiliser_objet (objet) ;           /* utiliser l'objet */
  }
}
```

FIG. 1 – Une solution du problème du producteur consommateur au moyen de sémaphores

Question 1.3 Les actions `Ai` ne soient jamais simultanées et se déroulent toujours dans l'ordre `A1`(`A2` ou `A3`)`A1`(`A2` ou `A3`)...

Question 1.4 Les actions `Ai` se déroulent toujours séquentiellement mais dans un ordre quelconque, par exemple : (`A2A1A3`)(`A2A3A1`)...

Exercice 2 (Examen de juin 1996)
Soient les deux processus `P1` et `P2` suivants. Ils se partagent deux sémaphores, `S1` et `S2` initialisés à 0.

```
P1 {                        P2 {
     procedure A1;               procedure A2;
     V(S2);                      V(S1);
     P(S1);                      P(S2);
     procedure B1;               procedure B2;
   }                           }
```

Question 2.1 Quelle synchronisation a-t-on imposée sur les exécutions des procédures `A1`, `A2`, `B1` et `B2` ?

Question 2.2 Écrire le code afin d'imposer la même synchronisation pour N processus en utilisant N sémaphores.

Question 2.3 On peut résoudre le problème pour N processus avec uniquement deux sémaphores (et un compteur). Donner le code des processus dans ce cas.

Exercice 3 (La mangeoire (d'après Tim Roberts)**)**
Un abri pour oiseaux comporte de nombreuses portes. Chaque porte possède un bouton à l'intérieur et un bouton à l'extérieur. L'abri peut contenir au maximum trois oiseaux.

Les oiseaux sont dressés pour appuyer sur les boutons s'ils veulent entrer ou sortir. Ils sont aussi dressés pour que un seul d'entre-eux ne rentre ou sorte quand une porte s'ouvre.

Quand un oiseau utilise un des boutons, les fonctions suivantes sont appelées :

```
int birds_in = 0 ;

bird_wants_to_leave (int button)
{
  birds_in-- ;
  open_door (button) ;
}
```

```
bird_wants_to_enter (int button)
{
  if (birds_in < 3) {
    birds_in++ ;
    open_door (button) ;
  }
}
```

Le code de ces fonctions est faux et pose de nombreux problèmes.

Question 3.1 Établir une liste des problèmes pouvant survenir.

Question 3.2 Utiliser des sémaphores pour résoudre ces problèmes.

Exercice 4 (La salle de bain (à nouveau d'après Tim Roberts)**)**
La salle de bain commune d'une pension mixte est utilisée par les filles et les garçons. Les règles suivantes sont observées :

1. au plus `MAX` filles utilisent simultanément la salle de bain ;
2. au plus `MAX` garçons utilisent simultanément la salle de bain ;
3. la salle de bain n'est jamais utilisé simultanément par des garçons et des filles ;

4. les filles ont priorité sur les garçons pour utiliser la salle de bain ;
5. les garçons ne peuvent entrer dans la salle de bain que si aucune fille n'utilise ou n'attend pour utiliser la salle de bain.

Question 4.1 À quelle condition une fille peut-elle entrer dans la salle de bain ? Et un garçon ?

Question 4.2 Identifier les événements qui peuvent bloquer un processus (fille ou garçon).

Question 4.3 Donner le code des quatre fonctions

```
void boy_wants_to_use_bathroom () ;
void boy_leaves_bathroom () ;
void girl_wants_to_use_bathroom () ;
void girl_leaves_bathroom () ;
```