

Systemes d'exploitation - Moniteurs II

Othmane AJDOR

2018-2019

Table des matières

1	But	3
2	Barrière à N	3
3	Allocateur	4
4	Producteurs Consommateurs	5

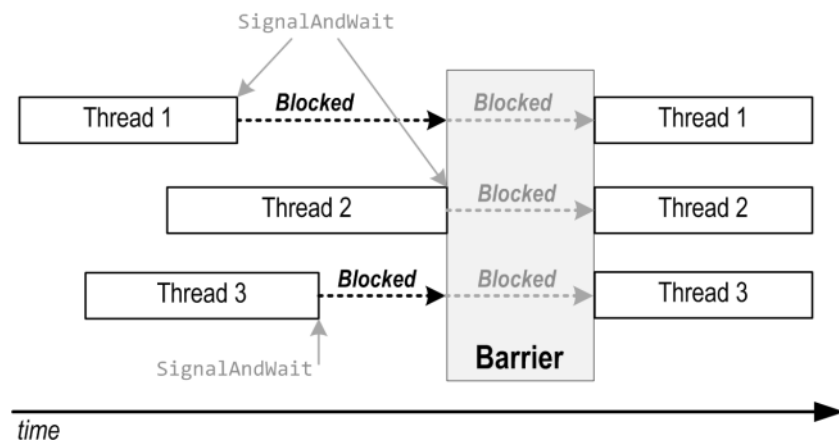
1 But

- Synchronisation de threads, 1 même programme , mémoire partagée
- Ensemble de fonctions en exclusion mutuelle (pas de problème de concurrence)

Le problème est de prendre des décisions alors qu'on est en exclusion mutuelle. On empêche le programme d'évoluer (mise en attente à l'aide de variables de condition), ainsi, on relâche l'exclusion mutuelle et les threads sont placés dans une file d'attente.

2 Barrière à N

On veut que N threads appellent `barriereN()`



On utilise un mutex[lock|unlock], la fonction `cnd_t[wait|signal]`.
`cnd_signal(&c)` réveille un thread.

```
int compteur = 0;
const N = 42;
mtx_t m;
cnd_t c;
barriereN(){
    mtx_lock();
    compteur++;
    while(compteur < N){
        cnd_wait(&c, &m);
    }
    // reveil en cascade, le N+1 reveil le premier et ainsi de suite
    cnd_signal(&c);
    mtx_unlock();
}
```

3 Allocateur

On se contente de reveiller tous les threads en attente.
cnd_broadcast(&c) reveille tous les threads.

```
const int resource = N;
mtx_t m;
cnd_t c;
allocation(int n){
    mtx_lock(&m);
    while(resource<n){
        cnd_wait(&c, &m);
    }
    resource -= n;
    mtx_unlock(&m);
}

liberation(int n){
    mtx_lock(&m);
    resource += n;
    cnd_broadcast(&c);
    mtx_unlock(&m);
}
```

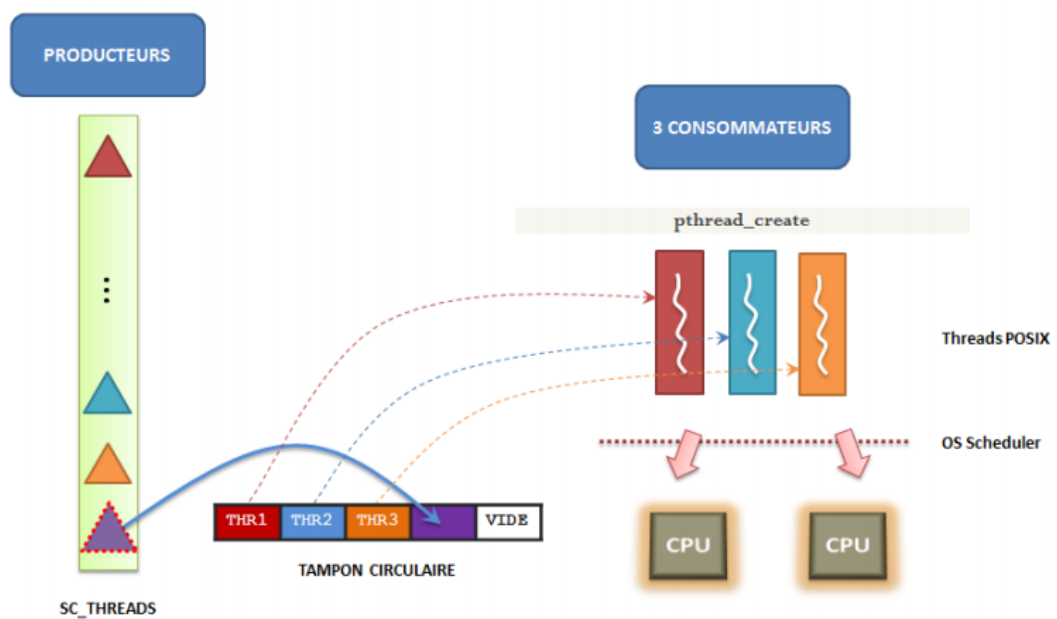
CODE CI-DESSOUS FAUX

Dans le cas où deux threads arrivent, ils se reveillent et s'endorment entre eux avec le code suivant :

```
const int resource = N;
mtx_t m;
cnd_t c;
allocation(int n){
    mtx_lock(&m);
    while(resource<n){
        cnd_wait(&c, &m);
        cnd_signal(&c);
    }
    resource -= n;
    mtx_unlock(&m);
}

liberation(int n){
    mtx_lock(&m);
    resource += n;
    cnd_signal(&c);
    mtx_unlock(&m);
}
```

4 Producteurs Consommateurs



```
int iecr = 0; // indice d'ecriture
int ilect = 0; // indice de lecture
int nb = 0;
mtx_t m;
cnd_t fp;
cnd_t fconso;
Msg Tab[N];

push(Msg msg){

}

Msg shift(){

}
```