

# Algorithmique Répartie

## Travaux Pratiques (2), Master 1 Informatique

### Communication

Il vous est demandé de spécifier en Promela, puis de les simuler, les algorithmes Producteur/Consommateur vus en cours. Vous considérerez chacun des cas suivants :

- 1 producteur / 1 consommateur, avec utilisation de deux (2) canaux. L'un pour véhiculer les messages du producteur au consommateur, et l'autre pour véhiculer les jetons du consommateur au producteur. Considérez le cas où la production des messages n'est pas séparée de leur émission. Donc, pas de tampon au niveau du producteur.
- 1 producteur / 1 consommateur, avec utilisation d'un (1) canal. L'unique canal permet de véhiculer à la fois les messages et les jetons. Simulez d'abord ce système en réduisant le nombre de canaux mais sans modifier les primitives d'émission/réception.
- M producteur / 1 consommateur. Généralisation du problème en considérant, comme dans le cours, les producteurs et le consommateur disposés en anneau. Au niveau d'un producteur, vous séparerez la production des messages de leur émission. Pour rappel, les primitives de l'algorithme vous sont données à la suite.

On considère un système composé de M Producteurs ( $Prod_0$  à  $Prod_{M-1}$ ) et 1 consommateur (Cons).

#### **/\* Variables du Producteur i \*/**

```
Tamponi[Ni]  
Ini init à 0  
Outi init à 0  
Nbmessi init à 0  
Nbauti init à 0
```

#### **/\* Variables du Consommateur \*/**

```
Tampon[N] ;  
In init à 0  
Out init à 0  
Nbmess init à 0  
Nblibere init à 0  
Present init à faux  
Prochain init à Prod0
```

### **/\* Partie Producteur \*/**

```
Produire(m)                               Sur_reception_de(j, (jeton, Nblib))
{  Attendre(Nbmessi < Ni) ;             {  tempi = Min(Nblib, Nbmessi - Nbauti) ;
  Tamponi[Ini] = m ;                   Nbauti += tempi ;
  Ini = (Ini + 1) % Ni ;               Nblib -= tempi ;
  Nbmessi++ ;                           Si (Nblib == 0) Alors
}                                           Envoyer_a(Cons, (jeton, Nblib)) ;
                                           Sinon
                                           Envoyer_a(Prod(i+1)%M, (jeton, Nblib)) ;
                                           Fsi
                                           }

Guetteuri()
{  Boucle Infinie
  Attendre(Nbauti > 0)
  Envoyer_a(Cons, Tamponi[Outi]) ;
  Outi = (Outi + 1) % Ni ;
  Nbmessi-- ;
  Nbauti-- ;
}
```

### **/\* Partie Consommateur \*/**

```
Consommer(m)                               Sur_reception_de(j, m)
{  Attendre(Nbmess > 0) ;                   {  Tampon[In] = m ;
  m = Tampon[Out] ;                           In = (In + 1) % N ;
  Out = (Out + 1) % N ;                       Nbmess++ ;
  Nbmess-- ;
  Nblibere++ ;
  Si (Present) Alors
    Envoyer_a(ProdProchain, (jeton, Nblibere))
    Nblibere = 0 ;
    Present = faux ;
  Fsi
}

Sur_reception_de(j, (jeton, Nblib))
{  Prochain = (j + 1) % M ;
  Nblib = Nblib + Nblibere ;
  Nblibere = 0 ;
  Si (Nblib > 0) Alors
    Envoyer_a(ProdProchain, (jeton, Nblib));
  Sinon
    Present = vrai ;
  Fsi
}
```