

TD 2 : Exclusion Mutuelle & Thread Java

1 Concepts

- A.1 A quoi servent les exclusions mutuelles ?
- A.2 Qu'est ce qu'une étreinte fatale ?
- A.3 Qu'est ce qu'une famine ?
- A.4 Donnez des exemples.

2 Diagramme d'état

- B.1 Construire le diagramme d'état pour l'algorithme ci-dessous à deux processus.
- B.2 Montrer sur le diagramme pourquoi l'exécution du programme ne satisfait pas l'exclusion mutuelle.

```
boolean occupe=false;

void Pmutex(){
    while (occupe)
        Thread.yield();
    occupe=true;
}

void Vmutex(){
    occupe=false;
}
```

3 Hyman 1966

Voici un mutex proposé par Hyman en 1966.

```
int turn=0;
boolean flag[]={false,false};

void Pmutex (int me){
    int you=1-me;

    1.     flag[me] = true ;
    2.     while (turn != me){
    3.         while (flag[you])
    4.             Thread.yield();
    5.         turn = me;
    }
}

void Vmutex(int me){
    6.     flag[me] = false ;
}
```

- C.1 Vérifie-t-il la propriété P (Progrès) ?
- C.2 Vérifie-t-il la propriété E (Équité) ?
- C.3 Vérifie-t-il la propriété EM (Sûreté) ?

Pour chaque réponse positive donnez les conditions nécessaires. Pour chaque réponses négative illustrez le avec un contre-exemple que vous testerez le avec l'algorithme de Peterson.

```
int turn=0;
boolean flag[]={false,false};

void Pmutex (int me){
    int you=1-me;
1.    flag[me] = true ;
2.    turn=you;
3.    while (flag[you] && turn == you){
4.        Thread.yield();
    }
}

void Vmutex(int me){
5.    flag[me] = false ;
}
}
```

4 Barrière (Le retour)

Voici une fonction de prise de barrière.

```
int nbThread=1;

synchronized void Attendre(){
    if (waiting<nbThread){
        waiting++;
        wait();
    }else{
        notifyAll();
        waiting=0;
    }
}
}
```

- D.1 Au vu du précédent TP, quelle partie doit-être atomique? (Si vous ne voyez pas demandez)
- D.2 Donnez un exemple de mauvais fonctionnement en cas de non atomicité.
- D.3 Quel mécanisme caché de java, permet-il à votre barrière de fonctionner correctement ?
- D.4 Transformez cette fonction afin de se passer de synchronized.