TD1 SE 2

Exercices

```
Tour=0
        while (1)
            { // attente active
            while (tour !=0) ;
            Section_critique() ;
            tour = 1 ;
            section_noncritique() ;
            ...
        }
// Processus P2
while (1)
        { // attente active
        while (tour !=1) ;
        Section_critique() ;
        tour = 0 ;
        Section_noncritique() ;
        ...
        }
```

Problème : On peut vérifier assez facilement que deux processus ne peuvent entrer en section critique en même temps, toutefois le problème n'est pas vraiment résolu car il est possible qu'un des deux processus ait plus souvent besoin d'entrer en section critique que l'autre; l'algorithme lui fera attendre son tour bien que la section critique ne soit pas utilisée.
Un processus peut être bloqué par un processus qui n'est pas en section critique.
P1 lit la valeur de tour qui vaut 0 et entre dans sa section critique.
Il est suspendu et P2 est exécuté. P2 teste la valeur de tour qui est toujours égale à 0. Il entre donc dans une boucle en attendant que tour prenne la valeur 1. Il est suspendu et P1 est élu de nouveau.
P1 quitte sa section critique, met tour à 1 et entame sa section non critique.
Il est suspendu et P2 est exécuté. P2 exécute rapidement sa section critique, tour = 0 et sa section non critique. Il teste tour qui vaut 0.
Il attend que tour prenne la valeur 1.

Exercice 2
Est-ce que ça marche

1 :

```
 Occupé  est un booleen := false
Entrer_sc() {
          While (occupé)
          Occupé=TRUE
}
Sortir_sc() {
          Occupé= FALSE
}
```

Non il y'a problème, si tous les deux arrivent en même temps, ils trouvent que c'est faux ils peuvent entrer en même temps à la section critique, et donc il y'a un problème comme vu au cours.

2 :

```
Int tour=0 ;
Bool je_veux_entrer [2]
Entrer_sc () {
        Je_veux_entrer[i] = TRUE
        Tour = i
        Attendre (je_veux_entrer[1 – i] == true et tour == 1 – i )
}
Sortir_sc () {
        Je_veux_entrer[i]=FALSE
}
```

Cette méthode marche pour deux processus, mais elle est mauvaise elle occupe le processeur à 100% pendant le quantum de temps du processus.