

## TD N° 2

### Exercice 1

1. Pourquoi le partage de ressources pose des problèmes dans un système multiprogrammé en temps partagé ?
2. Qu'est-ce qu'une section critique ?
3. Rappeler brièvement les exigences d'un bon mécanisme de contrôle d'accès à une section critique.
4. Soit les deux processus PA et PB qui partagent une variable k initialisée ainsi :

Int k=1;

Processus A	Processus B
<pre>int main (){     int i;     i=k;     i=i+1;     k=i;     printf("k=%d\n",k);     ... }</pre>	<pre>int main (){     int j;     j=k;     j=j+4;     k=j;     printf("k=%d\n",k);     ... }</pre>

Les résultats suivants sont-ils possibles ou non ?

	Processus A	Processus B
R1	K = 2	K = 2
R2	K = 2	K = 6
R3	K = 5	K = 2

3. On propose dans ce qui suit un algorithme pour une nouvelle version de l'alternance stricte.

```
void entrer_SC(int MonNumero)
{
    // MonNumero = 0 pour P0, 1 pour P1
    flag0 = true;
    while (flag1 == true); /* attente active */
}
void liberer_SC(int MonNumero)
{
    flag0 = false;
}
```

Cet algorithme respecte-t-il les 4 conditions précédemment citées ? Sinon, donner un exemple explicatif pour l'une des conditions violées.

---

## Exercice 2

---

Dans un système informatique, on dispose de trois fichiers F1, F2 et F3 et de trois processus dont les programmes

A, B et C ont les structures suivantes :

Programme A	Programme B	Programme C
Actions A1	Actions B1	Actions C1 (lire F3)
Actions A2 (lire F2)	Actions B2 (écrire F3)	Actions C2
Actions A3	Actions B3 (lire F1)	Actions C3
Actions A4 (écrire F3)	Actions B4	Actions C4 (écrire F2)
Actions A5	Actions A5	Actions C5

Chaque fichier ne peut ni être lu et modifié en même temps, ni modifié par plusieurs processus en même temps.

1. Donner pour chaque fichier les sections critiques de A, B et C.
2. En déduire les sections en exclusion mutuelle.

---

## Exercice 3

---

1. Rappeler le principe de l'algorithme de Peterson.
2. Prouver que cet algorithme respecte toutes les exigences d'un bon mécanisme de contrôle d'accès.