

## Corrigé type de l'examen

### Question de Cours : (7 pts)

Voir le support de cours.

### Exercice 1 :(8 pts)

On cherche à évaluer l'expression suivante :

$$e := (((b-d) * (a+c) + (e*f)) / (a+c))/(e*f)$$

1. Réaliser un découpage en tâches de cette expression sans l'ajout de variables intermédiaires. **(1,5 pt)**

$$t1: b = b - d$$

$$t2: a = a + c$$

$$t3: e = e * f$$

$$t4: b = b * a$$

$$t5: b = b + e$$

$$t6: b = b / a$$

$$t7: e = b / e$$

2. En vous servant de la définition de la condition de Bernstein, donner le graphe de précedence correspondant. **(2 pt)**

$$t1: \{R(t1) = d, W(t1) = b\};$$

$$t2: \{R(t2) = c, W(t2) = a\};$$

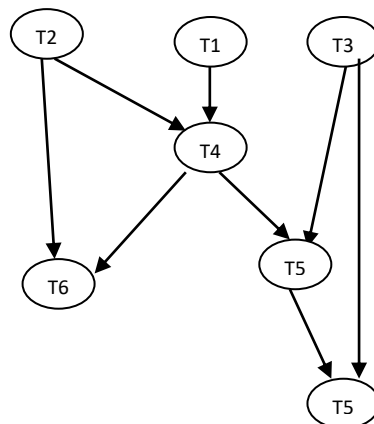
$$t3: \{R(t3) = f, W(t3) = e\}$$

$$t4: \{R(t4) = a, W(t4) = b\};$$

$$t5: \{R(t5) = e, W(t5) = b\};$$

$$t6: \{R(t6) = a, W(t6) = b\};$$

$$t7: \{R(t7) = b, W(t7) = e\}$$



3. En se basant sur le graphe obtenu, réaliser la synchronisation des tâches (processus) en utilisant quatre sémaphores S1, S2, S3 et S4. **(1.5 pt)**

Sémaphore S1=0, S2=0, S3=0, S4=0;

```
PRt1 (){  
t1: b = b - d;  
V(S1);  
}  
PRt2 (){  
t2: a = a + c;  
V(S1); V(S3);  
}  
PRt3 (){  
t3: e = e * f;  
V(S2);  
}  
PRt4 (){  
P(S1); P(S1);  
t4: b = b * a  
V(S2)  
}  
PRt5 (){  
P(S2);P(S2);  
t5: b = b + e;  
V(S3)  
}  
PRt6 (){  
P(S3);P(S3);  
t6: b = b / a;  
}  
PRt7 (){  
P(S4);P(S4);  
t7: e = b / e;  
}
```

#### 4. (3 pt)

```
Count1 =2, Count2 = 2, Count3 = 3, Count4 = 2;  
Begin  
S1;  
Fork L1;  
S2; Fork L2;  
S6; go to L3;  
L1: fork L4;  
S3; L2: joint count1;  
S7; fork L5;  
L3: joint count2; S8; go to L6;  
L4: fork L7;  
S4; go to L5;  
L7: S5;  
L5: joint count3;S9;  
L6: joint count4; S10;
```

## Exercice 2 : (5 pts)

Soit l'exécution parallèle des deux processus suivants :

```
var n : entier init 0 ; /* n entier initialisé à 0 */
Begin
Parbegin
Processus P1
Begin
n := n + 1;
End ;
Processus P2
Begin
n := n - 1;
End ;
Parend ;
End ;.
```

Notons l'instruction I1 c'est  $n := n + 1$  et l'instruction I2 c'est  $n := n - 1$ .

1. synchroniser les 2 processus de telle manière que les deux instructions se déroulent toujours dans l'ordre : I1I2I1I2I1I2I1I2... (2 pt)

Sémaphore  $s1=0$ , Sémaphore  $s2=0$  ;

ProcessusA :	ProcessusB :
Debut	Debut
Faire toujours	Faire toujours
P(s1) ;	P(s2) ;
<b>T1</b> ;	<b>T2</b> ;
V(s2) ;	V(s1) ;
Fait	Fait
Fin.	Fin.

2. Utilisez deux sémaphores pour synchroniser les 2 processus de telle manière que les deux instructions se déroulent toujours dans l'ordre : I1I1I2I2I1I1I2I2I1I1... (3 pt)

Sémaphore  $s1=2$ , Sémaphore  $s2=0$  ;

ProcessusA :	ProcessusB :
Debut	Debut
Faire toujours	Faire toujours
P(s1) ;	P(s2) ;
P(s1) ;	P(s2) ;
<b>T1</b> ;	<b>T2</b> ;
V(s2) ;	V(s1) ;
V(s2) ;	V(s1) ;
Fait	Fait
Fin.	Fin.

*Bon courage*