

Corrigé type

Exercice N° : 01 (06 pts : 02+ 02 + 02)

1. Lorsque le thread principal (ou le processus père) se termine, que se passe-t-il pour les threads secondaires (ou les processus fils) ?

Réponse : Lorsque le thread principal se termine, les threads secondaires s'arrêtent

Lorsque le processus père se termine, les processus fils continuent l'exécution

2. Quelles sont les signaux qui ne peuvent être ni capturés, ni ignorés ?

Réponse : SIGKILL, SIGSTOP

3. Comment peut-on éliminer la condition de « l'occupation et l'attente » afin de prévenir les interblocages ?

Réponse : Un processus ne doit demander des ressources supplémentaires qu'après avoir libéré les ressources qu'il occupe déjà.

Exercice N° : 02 (07 pts : 1.5 + 1.5 + 2 + 2)

A

- Valeurs possibles pour X et Y : X = 0, 1, 2 et Y = 0, 1, 2
- Une simple solution consiste à utiliser un seul sémaphore S initialisé à 1

Processus P1

Wait (S)

X := X + 1

Y := Y + 1

Signal (S)

Afficher (X, Y)

Processus P2

Wait (S)

X := X * 2

Y := Y * 2

Signal (S)

Afficher (X, Y)

B 1. A, ensuite B et C, ensuite D.../ Sémaphores : SA ← 1, SB, SC, SD ← 0

Processus_A

Wait (SA)

Afficher (A)

Signal (SB)

Signal (SC)

Processus_B

Wait (SB)

Afficher (B)

Signal (SD)

Processus_A

Wait (SC)

Afficher (A)

Signal (SD)

Processus_A

Wait (SD)

Wait (SD)

Afficher (A)

Signal (SA)

2 A, ensuite B ou exclusif C, ensuite D.../ Sémaphores SA ← 1, SBC, SD ← 0

Processus_A

Wait (SA)

Afficher (A)

Signal (SBC)

Processus_B

Wait (SBC)

Afficher (B)

Signal (SD)

Processus_A

Wait (SBC)

Afficher (A)

Signal (SD)

Processus_A

Wait (SD)

Afficher (A)

Signal (SA)

Exercice N° : 03 (07 pts : 2.5 + 1.5 + 03)

A.

1/ **Initialement, avant l'allocation de ressources => Need = Max**

On doit donc calculer les matrices de l'état Ti :

L'état actuel est décrit par :

$$\text{Available} = (1, 2, 1, 3) \text{ et } \text{Need} = \begin{pmatrix} 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 3 \\ 1 & 0 & 2 & 2 \end{pmatrix}$$

<P2, P3, P1> est une séquence saine => la situation est saine.

2/

$$R1(0, 1, 1, 0) \leq \text{Need}1(2, 2, 1, 1)$$

$$R1(0, 1, 1, 0) \leq \text{Available}(1, 2, 1, 3)$$

On suppose que le système accorde la requête :

L'état est décrit par :

$$\text{Available} = (1, 1, 0, 3), \quad \text{Need} = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 3 \\ 1 & 0 & 2 & 2 \end{pmatrix}$$

$$\text{Allocation} = \begin{pmatrix} 2 & 2 & 3 & 0 \\ 1 & 1 & 2 & 1 \\ 1 & 2 & 1 & 1 \end{pmatrix}$$

Cet état est malsain car il n'existe pas de séquence saine, donc on doit restaurer l'ancien état et P1 doit attendre.

B.

	<i>Allocation</i>			<i>Max</i>			<i>Need</i>			<i>Available</i>		
	C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_1	C_2
P_0	1	0	0	1	1	0	0	1	0	0	0	0
P_1	0	1	0	0	1	1	0	0	1			
P_2	0	0	1	1	0	1	1	0	0			