

ETCD (durée : une heure)

Exercice 1 :

- Donnez la solution de Peterson vue (en cours) pour le problème de la section critique.
- Cette solution vérifie-t-elle la condition de progression ?

Exercice2

Soit P1 et P2 deux processus parallèles. Les algorithmes de ces deux processus sont écrits comme suit:

<u>Processus P1 :</u>	<u>Processus P2 :</u>
Debut	Debut
Répéter	Répéter
T1 ;	T2 ;
Jusqu'à faux	Jusqu'à faux
Fin.	Fin.

- Utilisez un sémaphore pour synchroniser les deux processus de telle manière que l'exécution de la tâche T1 ne soit jamais simultanée avec l'exécution de la tâche T2.
- Utilisez deux sémaphores pour synchroniser les deux processus de telle manière que les tâches se déroulent toujours dans l'ordre : T1T2T1T2T1T2...
- Utilisez deux sémaphores pour synchroniser les deux processus de telle manière que les tâches se déroulent toujours dans l'ordre : T1T2T2T1T2T2T1T2T2...

Exercice3

Soient les trois processus P1, P2 et P3 suivants qui partagent deux sémaphores, SA et SB initialisés à 0, 0.

<u>Processus P1 :</u>	<u>Processus P2 :</u>	<u>Processus P3 :</u>
Debut	Debut	Debut
T1 ;	T2 ;	T3 ;
P(SA) ;	V(SA) ;	P(SB) ;
T4 ;	V(SB) ;	T5 ;
Fin.	Fin.	Fin.

- Quelle synchronisation a-t-on imposée sur les exécutions des tâches T1, T2, T3, T4, T5
- Donner le graphe de précedence.

Corrigé - contrôle continu SE2 (2022-2023)

Exo: 1

- a. La description de l'algorithme de Peterson: voir support de cours.
- b. La condition de progression est définie par le fait qu'un processus qui opère en dehors de la section critique (S.C.), ne doit pas empêcher un autre processus d'y accéder.

→ supposons que P_i est en dehors de la section critique \Rightarrow

Drapeau $[i] = \text{faux}$.

Si P_j demande la S.C. \Rightarrow il met Drapeau $[j]$ à vrai et tour à i ,
 P_j ensuite vérifie la condition de l'attente active, il la trouve à faux
car Drapeau $[i]$ est à faux $\Rightarrow P_j$ franchira la barrière avec
succès et accède à la S.C. \Rightarrow progression assurée.

Exo: 02

- a) T_1 et T_2 ne soient jamais simultanées \Rightarrow Exclusion mutuelle \neq

utilisation d'un sémaphore binaire:

Sémaphore $\text{mutex} = 1;$

P_1 Début $P(\text{mutex});$ $T_1;$ $V(\text{mutex});$ fin.	P_2 Début $P(\text{mutex});$ $T_2;$ $V(\text{mutex});$ fin.
--	--

EAO:02 (suite).

b. T_1 doit s'exécuter tjrs avant T_2
 \Rightarrow utilisation de deux sémaphores privés.

Sol:

Sémaphores S_1, S_2 init à 0,0;

P₁

Debut

$T_1;$
 $V(S_2);$
 $P(S_1);$

fin.

P₂

Debut

$P(S_2);$
 $T_2;$
 $V(S_1);$

fin.

c. $T_1 T_2 T_2 T_1 T_2 T_2 \dots$

Sol:

sémaphores S_1, S_2 init à 0,0;

Processus P₁

Debut

$T_1;$
 $V(S_1); V(S_2);$
 $P(S_1); P(S_2);$

fin.

Processus P₂

Debut

$P(S_2);$
 $T_2;$
 $V(S_1);$

fin.

EAO:03

a. La synchronisation imposée sur les exécutions des tâches T_1, T_2, T_3, T_4 est la suivante.

- 1] T_1, T_2, T_3 sont indépendantes
- 2] T_4 doit attendre T_1 et T_2 (T_4 est une tâche de rendez-vous entre T_1 et T_2);
- 3] T_5 doit attendre l'exécution de T_2 et T_3 (T_5 : tâche de R.D.V entre T_2 et T_3).

b. graphe de précedence.

