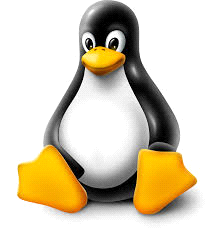


Compte Rendu De Travaux Pratique

System d’Exploitation

Travaille Pratique 6 : Les Tubes



SMI – S4

Réaliser par :

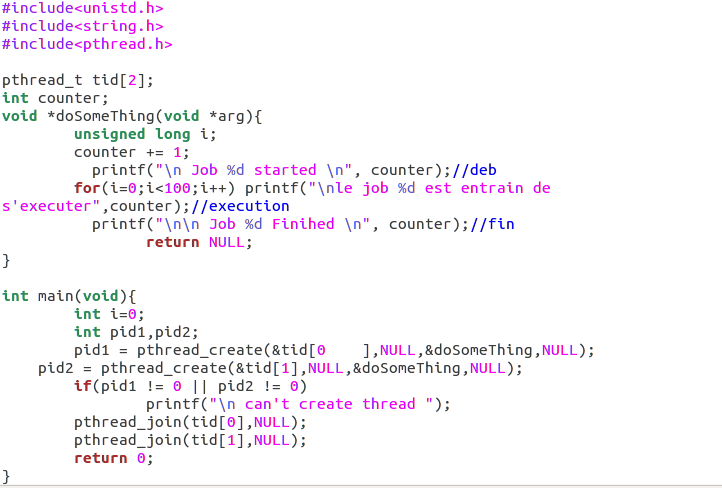
**Prénom :** **CNE :** **Nom :**

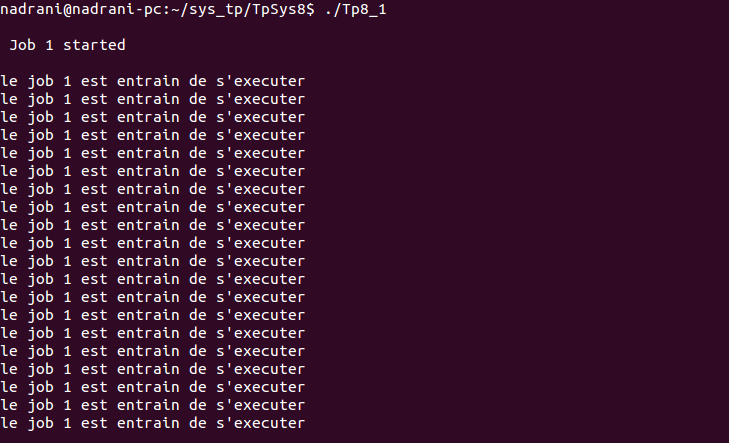
Achraf 1513755449 Tazi

Oussama 1311778906 Nadrani

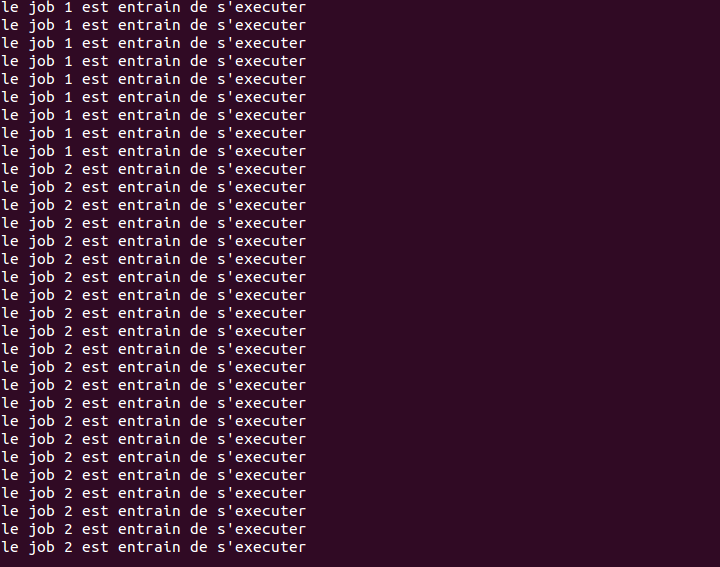
***Exercice1 :***

***Ecrire un programme qui crée deux threads dont chacune appelle la même fonction nommée ‘doSomeThing’.Cette dernière imprime au début de son exécution un message du commencement de l’exécution ainsi que la valeur de la variable ‘counter’ qui correspond à un compteur. Après cette impression, la fonction effectue un certain traitement et imprime la fin de son exécution par un message accompagné de la valeur du ‘counter’. Les deux fonctions partagent la variable ‘counter’. Cette variable s’incrémente chaque fois que ‘doSomeThing’ est appelé par l’un ou l’autre des deux threads. Que remarquez vous ?***

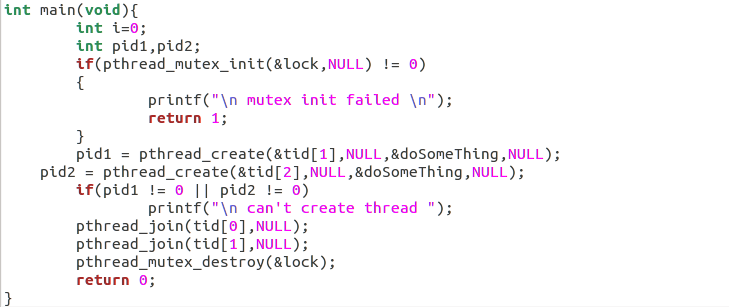
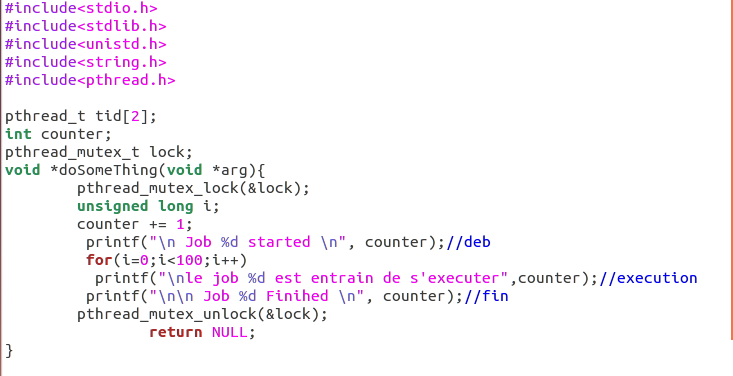
****

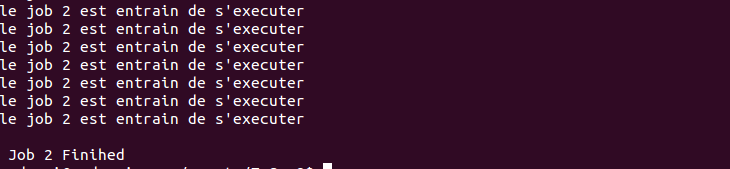
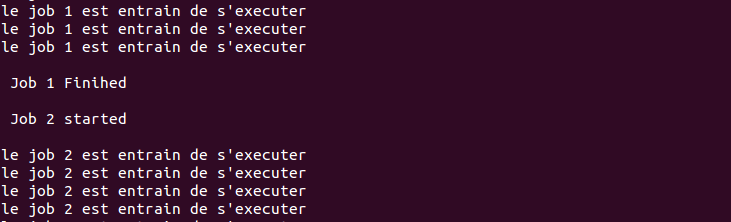
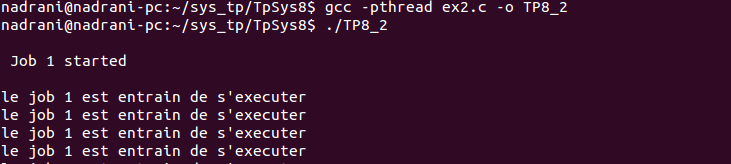
**Résultat :**

**Remarque :**

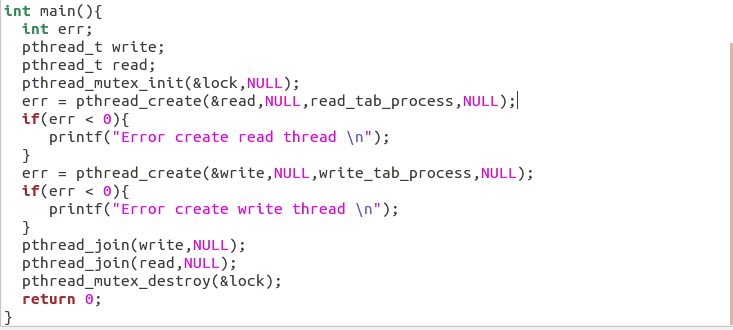
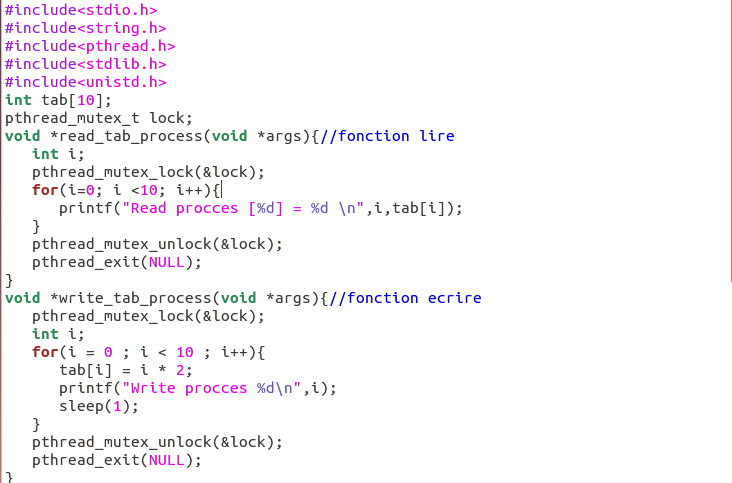
On remarque que la variable globale **counter** a était incrémentais avant la fin du premier thread

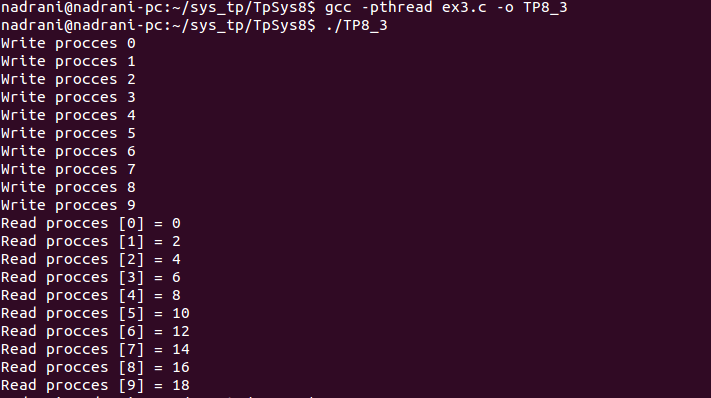
***Exercice2 :***

***Modifier le programme de l’exercice 1 en introduisant les fonctions de synchronisation les mutex afin d’afficher les bons résultats.***

**Résultat :**

***Exercice3 :***

***En utilisant les fonctions de synchronisation de mutex, écrire un programme C qui crée deux threads qui appellent deux fonctions (a) ‘write\_tab\_process’ et (b) ‘read\_tab\_process’. La 1ère fonction (a) remplie un tableau tab de 10 éléments dont chaque case contient le double de son indice (2\*i). La 2ème fonction (b) affiche le contenu de tab sur la sortie standard. Il est à préciser que les deux threads partagent entre elles le tableau tab. ***

**Résultat :****