Exercice 01

Le parallélisme est pour

1. Donner une vision de programmation clair Faux
2. Les limites de l'approche microprocesseur Vrai

La limite des machine Von Newman en term de vitesse de traitement a poussé les chercheurs de proposer d'autres solutions où le parallélisme un des ces solutions.

3. Pour accélérer les sites Webs Faux
4. A cause de l'existence de la propriété du parallélisme dans les applications. Vrai

Plusieurs applications et domaines supportent le parallélisme comme le domaine de traitement d'images

5. Amélioration des performances de calcul Vrai

L'un des objectif d'executer les programme en parallèle est d'augmenter la performance et principalement en term de temps d' execution.

6. Traiter les données volumineuses Vrai

Avec l'explosion de données numérique, le parallélisme joue un role important pour accélérer et augmenter la performance.

Exercice 02

1. Quelle est la relation entre le cout, le temps et le parallélisme

Le parallélisme joue un role important dans nombreux domaines mais il y a une relation entre le parallélisme et cout. Le but principal d'implémenter une architecture parallèle est pour minimiser le temps de calcul mais on doit toujours prendre en considération le cout nécessaire pour acheter une arch. Parall.

2. Le parallélisme joue un rôle important dans de nombreux domaines. Donner trois (03) domaines nécessite le parallelism pour améliorer le calcule.

Traitement d'image, Big data et Médecine

3. Schématiser la classification de Flynn

Voir le cours

4. Selon la classification de Flynn, un système distribué rentre dans quelle classe?

MIMD

5. Pour Accéder au calculateur Ibnkhaldoune on doit taper une commande de le console. Parmi les commandes suivantes donner la commande correcte:

ssh user@IP -p Port

Exercice 03

Soit le code suivant:

```
from multiprocessing import Process, Queue
import time
def fun(name, wait, q):
      time.sleep(wait)
      counter = 0
      print(f'Hello {name}')
      while wait:
            q.put(name)
            print(f'put {name at: }', wait)
            counter = counter + 1
```

```
            wait = wait - 1
            time.sleep(wait)
      print('finish with', counter)
def main():
      q = Queue()
      p1 = Process(target=fun, args=('Work Home', 5,q))
      p2 = Process(target=fun, args=('Exam', 3,q))
      p1.start()
      p2.start()
      p1.join()
      p2.join()
if __name__ == '__main__':
      main()
```

1. Remplir la table ci-dessous après l'execution du code
2. Est ce que il y a des moments où les deux processus fair simultanément un put dans le Queue. Si oui, donner les temps de ces moments et proposer une solution pour éviter cette action.

Oui, voir la ligne 5 de la table p1 et p2. On peut utiliser Lock pour éviter cette situation.

Time p1	Print	Queue	counter
0			
1			
2			
3			
4			
5	Hello Work Home Put Work Home at: 5	Exam (Exam, Work Home)	0 1
6			
7			
8			
9	Put Work Home at: 4	Exam | (Exam, Work Home) | Work Home	1 2
10			
11			
12	Put Work Home at: 3	Exam | (Exam, Work Home) | Work Home | Work Home	2 3
13			
14	Put Work Home at: 2	Exam | (Exam, Work Home) | Work Home | Work Home | Work Home	3 4
15	Put Work Home at: 1	Exam | (Exam, Work Home) | Work Home | Work Home | Work Home | Work Home	4 5
16			

Time p2	Print	Queue	counter
0			
1			
2			
3	Hello Exam put Exam at: 3	Exam	0 1
4			
5	put Exam at: 2	Exam (Exam, Work Home)	1 2
6	put Exam at: 1		2 3