

TD: composants intégrés d'un système à puce et processus:

Exercice 1: Vrai - Faux:

1. Un problème crucial pour un SoC est la grande consommation d'énergie.
2. La finesse de gravure d'un SoC est de l'ordre de $10^{-6}m$.
3. La miniaturisation des circuits peut permettre des gains de performance.
4. Avec deux processeurs de même fréquence, les vitesses de traitement sont égales ou très proches.
5. Un processus peut être enregistré sur un disque ou une clé USB.
6. Un processus a le même PID à chaque session.
7. Pour la lecture d'une vidéo, le GPU est plus utilisé que le CPU.
8. Il y a autant de processus que de programmes.
9. Un processus actif peut passer à l'état bloqué ou à l'état prêt.
10. Les phénomènes d'interblocage concernent un processus et son processus parent.

Exercice 2:

Si dans les années 1970, on pouvait placer 2000 transistors sur une surface de $10mm^2$, quelle surface aurait été nécessaire pour placer 2 milliards de transistors?

Exercice 3:

Sur l'image d'un SoC, on peut lire Adreno 630, Hexagon 685, Kryo 385, X20 LTE, Spectra 280. À quels composants du SoC correspondent ces dénominations?

Exercice 4:

1. Ouvrir un terminal et entrer la commande `ps - aef`. Noter le PID du processus `ps - aef`.
2. Quel est le processus parent du processus `ps - aef` et quel est son PID?
3. Quel est le processus parent du processus parent du processus `ps - aef` et quel est son PID?
4. Fermer le terminal, le rouvrir et entrer à nouveau la commande `ps - aef`. Quelles différences peut-on noter par rapport aux résultats des questions précédentes?

Exercice 5:

1. Ouvrir firefox et déterminer combien de processus sont créés.
2. Quels sont les PID de chaque processus?
3. Que se passe-t-il si depuis la fenêtre de firefox, on ouvre un nouvel onglet, ou si on ouvre une nouvelle fenêtre?

Exercice 6:

On s'intéresse à la situation suivante:

- Un processus P1 mobilise une ressource R1 pendant qu'un deuxième processus P2 mobilise une ressource R2.
- P1 attend la ressource R2 avant de libérer R1 et P2 attend la ressource R1 avant de libérer R2.

1. Schématiser la situation précédente.
2. A quel phénomène a-t-on affaire ici? Que doit-on faire pour débloquer la situation?
3. trouver une analogie avec le programme suivant:

```
0 while a == 0 or b == 1:  
1     if b != 1:  
2         a = 1  
3     if a != 0:  
4         b = 0
```

interblocage.py

Exercice 7:

On considère 3 processus P1, P2 et P3 tels que:

- P1 demande R1, demande R2, libère R1, libère R2,
- P2 demande R2, demande R3, libère R2, libère R3,
- P3 demande R3, demande R1, libère R3, libère R1,

1. Y a-t-il interblocage si les processus sont exécutés successivement: P1 puis P2 puis P3?
2. Décrire une exécution des 3 processus qui conduit à une situation d'interblocage. La schématiser.