Université Mohamed Khider Biskra
Faculté des sciences exactes et des sciences de la nature et de la vie
Département d'informatique

Niveau: M1 (tous les parcours) **Module: SD**

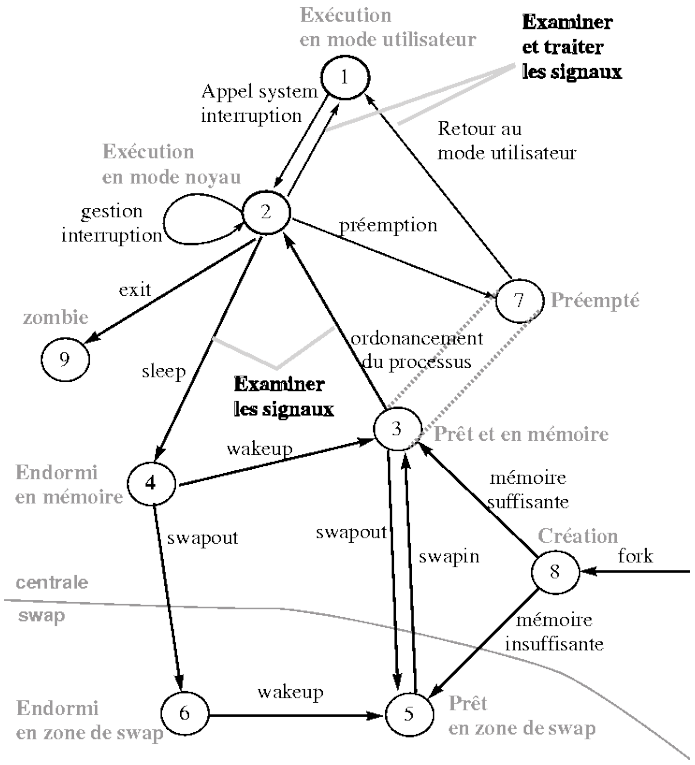
Année: 2023-2024

Série d'exercices n°1

Introduction aux systèmes répartis
Rappel sur les systèmes d'exploitation centralisés
Problème d'ordonnancement et de synchronisation

Questions

1. Expliquer le schéma suivant représentant les états possibles d'un processus et les transitions entre ces états. Expliquer pourquoi certaines transitions ne sont pas possibles.



2. Qu'est-ce qu'un système distribué ? Quel est l'objectif principal des systèmes distribués ?
3. Donner les différences entre un système d'exploitation centralisé et système d'exploitation distribué ? Quel est le principal avantage des systèmes distribués par rapport aux systèmes centralisés ?
4. Lequel des éléments suivants n'est PAS un défi dans les systèmes distribués : a. Latence du réseau et largeur de bande b. Évolutivité c. Sécurité réduite ? Expliquer ? Quel est le théorème CAP cohérence (Coherence), disponibilité (Availability) et tolérance au partitionnement (Partition tolerence) dans les systèmes distribués ? Expliquer ?
5. Quelle est la différence entre la communication synchrone et asynchrone dans les systèmes distribués ? Quel modèle de communication est généralement utilisé dans les systèmes distribués pour la communication inter-processus ? Expliquer en détail ce modèle, ensuite justifier son utilisation.
6. Quelle est la différence entre un système distribué et un système parallèle ?
7. Citez deux architectures courantes pour les systèmes distribués, expliquer en détails.
8. Donner lé définition d'un middleware dans un système distribué. Décrire à l'aide de schéma son fonctionnement. Donner les avantages en tant que couche intermédiaire de logiciel située entre l'application et le réseau.

Exercice 1

1. Expliquer l'algorithme d'ordonnancement « Le plus court d'abord ».
2. Considérons cinq travaux A, B, C, D et E, dont les temps d'exécution et leurs arrivages respectifs sont donnés dans la table suivante. Faire un schéma qui illustre son exécution et calculer le temps de séjour de chaque processus, le temps moyen de séjour, le temps d'attente et le temps moyen d'attente en utilisant : 1. FIFO 2. Le plus court d'abord.

Processus	Temps d'exécution	Temps d'arrivage
A	3	0
B	6	1
C	4	4
D	2	6
E	1	7

Exercice 2

1. Donner le mode de fonction de chacun des algorithmes suivants :
    a. Le plus petit temps de séjour ou Shortest Remaining Time SRT.
    b. L'algorithme du tourniquet, circulaire ou Round robin.
2. Soient deux processus A et B prêts tels que A est arrivé en premier suivi de B, 2 unités de temps après. Les temps de UCT nécessaires pour l'exécution des processus A et B sont respectivement 15 et 4 unités de temps. Le temps de commutation est supposé nul. Calculer le temps de séjour de chaque processus A et B, le temps moyen de séjour, le temps d'attente, le temps moyen d'attente, et le nombre de changements de contexte pour :
    a. SRT,
    b. Round robin (quantum = 10 unités de temps),
    c. Round robin (quantum = 3 unités de temps).

Exercice 3

On dispose de processus a, b, c, et init qui sont lancés au même instant. On désire contrôler l'ordonnancement des actions des processus a, b, c. Pour cela on dispose d'un langage de spécification comportant les opérations de concaténation séquentielle * et ., de choix exclusif + et de mise en parallèle ||. Les processus a, b, c exécutent le code suivant :

```
Processus x() /* les trois codes sont différent pour x= a, b ou c*/
Boucle (avantx ; actionx ; apresx ;)
Fin ; /* actionx n'est pas forcément atomique*/
```

On note que si avantx et apresx sont vides pour un processus x donné, alors celui ci exécute indéfiniment actionx, de façon séquentielle mais parallèlement aux autres processus. Donner, pour les spécifications suivantes :

(a.b)*, (a.(b || c))*, (a.(b+c))*, (a || b)* le code (qui est toujours une séquence de P et/ou V) de init, avantx, et apresx pour chaque processus a, b. c.