Commencé le	dimanche 28 avril 2024, 22:46
État	Terminé
Terminé le	dimanche 28 avril 2024, 22:49
Temps mis	3 min 22 s
Note	**10,00** sur 10,00 (**100**%)

Question 1

Correct

Note de 1,00 sur 1,00

L'ordonnanceur d'un système d'exploitation de la famille Unix, Mac ou Windows se charge de gérer l'allocation du (des) processeur(s) aux processus prêts. Le temps alloué à chaque processus est réparti par le processus entre ses threads prêts.

Veuillez choisir une réponse.

- Vrai
- Faux ✓

La réponse correcte est « Faux ».

Question 2

Correct

Note de 1,00 sur 1,00

Dans un ordonnancement non-préemptif de threads, un thread bascule de l'état en exécution vers l'état prêt uniquement dans le cas où il cède le processeur.

Veuillez choisir une réponse.

- Vrai ✓
- Faux

La réponse correcte est « Vrai ».

Question 3

Correct

Note de 1,00 sur 1,00

Supposez un système qui gère l'exécution des threads selon un ordonnancement préemptif à base de priorités fixes.

L'état du thread en exécution bascule vers l'état prêt si . ...

Sélectionnez les conditions qui complètent correctement la phrase précédente.

Veuillez choisir au moins une réponse.

- a. le thread a consommé beaucoup de temps CPU.
- b. un thread plus prioritaire arrive dans le système ou devient prêt. ✓
- c. le thread cède le processeur. ✓
- d. le thread provoque un défaut de page dur.

Votre réponse est correcte.

Les réponses correctes sont : un thread plus prioritaire arrive dans le système ou devient prêt., le thread cède le processeur.

Question 4

Correct

Note de 1,00 sur 1,00

Supposez un système qui gère l'exécution des threads selon un ordonnancement circulaire.

L'état du thread en exécution bascule vers l'état prêt si ...

Sélectionnez les conditions qui complètent correctement la phrase précédente.

Veuillez choisir au moins une réponse.

- a. le thread a consommé son quantum et il y a au moins un autre thread à l'état prêt. ✓
- b. un thread arrive dans le système ou devient prêt.
- c. le thread a consommé son quantum et il n'y a aucun autre thread à l'état prêt.
- d. le thread cède le processeur. ✓

Votre réponse est correcte.

Les réponses correctes sont :
le thread a consommé son quantum et il y a au moins un autre thread à l'état prêt.,

le thread cède le processeur.

Question 5

Correct

Note de 1,00 sur 1,00

Supposez un système qui gère l'exécution des threads selon un ordonnancement préemptif à files multiples avec 32 niveaux de priorités. L'ordonnancement des threads prêts qui partagent la même priorité est circulaire avec un quantum de 10ms.

L'état du thread en exécution bascule vers l'état prêt si . ...

Sélectionnez les conditions qui complètent correctement la phrase précédente.

Veuillez choisir au moins une réponse.

- [x] a. le thread a consommé 10ms de temps CPU et il y a au moins un thread prêt de même priorité. ✓
- [x] b. un thread plus prioritaire arrive dans le système ou devient prêt. ✓
- [ ] c. le thread exécute l'équivalent de l'instruction (V(mutex)).
- [ ] d. le thread a consommé 10ms de temps CPU et il y a au moins un thread prêt de priorité inférieure à celle du thread en exécution.

Votre réponse est correcte.

Les réponses correctes sont :
le thread a consommé 10ms de temps CPU et il y a au moins un thread prêt de même priorité. ,

un thread plus prioritaire arrive dans le système ou devient prêt.

Question 6

Correct

Note de 1,00 sur 1,00

Dans un système, où l'ordonnancement est préemptif à base de priorités, deux tâches T1 et T2 concurrentes partagent en exclusion mutuelle une ressource R. La tâche T1 est plus prioritaire que la tâche T2. Dans un tel système, il y a un risque d'**inversion de priorités** (la tâche T2 détient la ressource R et T1 est en attente de R). L'utilisation du protocole PIP *(Priority Inheritance Protocol) permet de ...*

Veuillez choisir une réponse.

- [ ] a. éliminer tout risque d'inversion de priorités.
- [ ] b. éliminer tout risque d'interblocage.
- [x] c. borner la durée de l'inversion de priorités. ✓

Votre réponse est correcte.

La réponse correcte est :
borner la durée de l'inversion de priorités.

Question 7

Correct

Note de 1,00 sur 1,00

Avec le protocole PIP, *la priorité de T2 devient égale à celle de T1, lorsque T1 demande la ressource R. T2 reprendra sa priorité lorsqu'il libérera la ressource.*

Veuillez choisir une réponse.

- Vrai ✓
- Faux

La réponse correcte est « Vrai ».

Question 8

Correct

Note de 1,00 sur 1,00

Contrairement aux sémaphores POSIX, les mutex POSIX offrent la possibilité de traiter les inversions de priorités.

Veuillez choisir une réponse.

- Vrai ✓
- Faux

Il est possible de spécifier lors de l'initialisation d'un mutex s'il y a lieu d'utiliser un protocole pour traiter l'inversion des priorités :

Protocol is an optional attribute for Pthread mutexes. The values for mutex protocol are:

- PTHREAD_PRIO_NONE – Default. Thread priorities are not modified by locking the mutex.
- PTHREAD_PRIO_INHERIT – Use the priority inheritance protocol.
- PTHREAD_PRIO_PROTECT – Use the priority ceiling protocol.

(man pthread pour plus d'info).

La réponse correcte est « Vrai ».

Question 9

Correct

Note de 1,00 sur 1,00

Dans certains systèmes d'exploitation, l'ordonnancement des threads est préemptif et à files multiples. Ils associent une priorité de base et une priorité dynamique à chaque thread créé.

Sélectionnez les énoncés qui pourraient contribuer à atteindre à favoriser les threads interactifs qui font d'E/S et peu de traitement.

Veuillez choisir au moins une réponse.

- [x] a. La priorité dynamique augmente lorsque le thread est en attente (prêt ou bloqué). ✓
- [ ] b. La priorité dynamique augmente lorsque le thread est en exécution.
- [x] c. La priorité dynamique diminue lorsque le thread est en exécution. ✓
- [ ] d. La priorité dynamique diminue lorsque le thread est en attente.

Votre réponse est correcte.

Les threads/processus interactifs font beaucoup d'E/S. Ils vont se retrouver souvent en attente d'E/S.

Les réponses correctes sont :
La priorité dynamique augmente lorsque le thread est en attente (prêt ou bloqué).,

La priorité dynamique diminue lorsque le thread est en exécution.

Question 10

Correct

Note de 1,00 sur 1,00

Soit S = {T1(C1,D1,P1), ..., Tn(Cn,Dn,Pn)} un ensemble de tâches temps réel périodiques et indépendantes. Les tâches de S sont ordonnançables (il existe une politique d'ordonnancement qui permet aux tâches de respecter leurs échéances) si $C1/D1 + ...+ Cn/Dn \leq 1$.

Veuillez choisir une réponse.

- (•) Vrai ✓
- ( ) Faux

Pour des tâches périodiques indépendantes, EDF nous offre une condition suffisante d'ordonnancement qui est : $C1/D1 + ...+ Cn/Dn \leq 1$.

La réponse correcte est « Vrai ».