

Commencé le	dimanche 2 avril 2023, 08:50
État	Terminé
Terminé le	dimanche 2 avril 2023, 08:52
Temps mis	1 min 42 s
Note	30,00 sur 10,00 (300%)
Dans l'évaluation avec indication de certitude, la note ci-dessus est affichée relativement au maximum pour toutes les réponses correctes, avec C = 1. ?	
Résultats tout le test (10 questions)	
Évaluation moyenne du degré de certitude	3,00
Précision	100,0%
Bonus degré de certitude	0,0%
Précision + bonus	100,0%
Ventilé par certitude	
C = 3	Réponses : 10. Précision : 100% . (Plage optimale 80% à 100%). Vous étiez OK en sélectionnant ce degré de certitude.
C = 2	Pas de réponse
C = 1	Pas de réponse

Question 1

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Un thread:

Veuillez choisir au moins une réponse.

- ☒ 1. Possède sa propre pile d'exécution ✓ Oui, chaque thread possède sa propre pile d'exécution.
- ☒ 2. Partage le même espace de mémoire avec tous les threads composant son processus ✓ Oui, il y a partage mémoire entre tous les threads d'un processus.
- ☐ 3. S'exécute sur plusieurs coeurs ou processeurs, simultanément
- ☐ 4. Partage le même espace de mémoire avec tous les processus composant son thread
- ☐ 5. Est composé de plusieurs processus

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

Les réponses correctes sont : Partage le même espace de mémoire avec tous les threads composant son processus, Possède sa propre pile d'exécution

Question 2

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Le temps nécessaire pour créer un thread dans un processus existant est:

Veuillez choisir une réponse.

- ☐ 1. Plus grand que le temps nécessaire pour créer un nouveau processus
- ☒ 2. Plus petit que le temps nécessaire pour créer un nouveau processus ✓ Oui
- ☐ 3. Équivalent au temps requis pour créer un nouveau processus

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

La réponse correcte est : Plus petit que le temps nécessaire pour créer un nouveau processus

Question 3

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Si deux threads écrivent la même variable entière sans synchronisation avec deux valeurs différentes, le résultat sera:

Veuillez choisir une réponse.

- ☐ 1. La valeur écrite par le premier thread créé (celui qui aura le tid le plus petit)
- ☒ 2. Une des deux écritures, mais sans pouvoir déterminer laquelle ✓ Oui !
- ☐ 3. Une valeur aléatoire autre que celles écrites
- ☐ 4. Un mélange des deux valeurs, chaque octet venant aléatoirement d'une des deux valeurs écrites

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

La réponse correcte est : Une des deux écritures, mais sans pouvoir déterminer laquelle

Question 4

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Juste après un appel à `exec`, quel(s) élément(s) suivant(s) peuvent changer pour le processus ?

Veuillez choisir au moins une réponse.

- ☒ 1. Les fichiers ouverts ✓ Oui
- ☒ 2. Le programme exécuté ✓ Oui
- ☐ 3. Le PID
- ☒ 4. Les variables locales ✓ Oui

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

Les réponses correctes sont : Les fichiers ouverts, Le programme exécuté, Les variables locales

Question 5

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Juste après un `fork`, quel(s) élément(s) suivant(s) peuvent différer entre le processus père et le processus fils ?

Veuillez choisir au moins une réponse.

- ☐ 1. Les fichiers ouverts
- ☒ 2. Le PID ✓ Le PID est différent dans le processus fils (car on a créé un nouveau processus par copie du père). Donc tout le reste est identique au père (juste après le `fork`), sauf le PID. Donc en fonction de la valeur de retour du `fork` on saura si on est dans le père ou le fils et on exécutera un code différent la plupart du temps.
- ☐ 3. Les variables locales
- ☐ 4. Le programme exécuté

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

La réponse correcte est : Le PID

Question 6

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

En considérant le morceau de code suivant:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int main() {
    int i=0;
    pid_t father_pid = getpid();
    if (fork()) {
        i++;
    }
    i++;
    printf("%s: i=%d\n", (father_pid == getpid()) ? "PP" : "F ", i);
    wait(NULL);
}
```

Quelles sont les valeurs de *i* affichées par le processus père (P) et par le processus fils (F):

Veuillez choisir une réponse.

- ☐ a. P: i = 1 et F: i = 1
- ☐ b. P: i = 1 et F: i = 2
- ☐ c. P: i = 2 et F: i = 2
- ☒ d. P: i = 2 et F: i = 1 ✓

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

La réponse correcte est : P: i = 2 et F: i = 1

Question 7

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Un processus zombie est un processus qui:

Veuillez choisir une réponse.

- ☒ 1. A terminé son exécution et attend la prise en compte de cette fin par son père ✓ Oui, c'est la bonne définition d'un processus zombie.
- ☐ 2. A terminé son exécution en erreur
- ☐ 3. A perdu son père et a été adopté par le processus init
- ☐ 4. A perdu son père et n'est plus rattaché à l'arborescence des processus

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

La réponse correcte est : A terminé son exécution et attend la prise en compte de cette fin par son père

Question 8

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

La mise en œuvre des threads au niveau du système d'exploitation peut se faire:

Veuillez choisir au moins une réponse.

- ☒ 1. En espace noyau (Kernel Space) ✓ Oui
- ☐ 2. Dans la pile d'exécution
- ☒ 3. En espace utilisateur (User Space) ✓ Oui

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

Les réponses correctes sont : En espace utilisateur (User Space), En espace noyau (Kernel Space)

Question 9

Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Quelle(s) affirmation(s) sont vraie(s)?

Veuillez choisir au moins une réponse.

- ☐ 1. Le processus parent partage son espace d'adressage avec son processus enfant
- ☒ 2. Le processus parent connaît le processus enfant qu'il a engendré ✓ Oui grâce à la valeur de retour de la fonction fork dans le père.
- ☒ 3. Un processus peut connaître son processus parent ✓ Oui grâce à la fonction getppid.
- ☒ 4. Le processus parent peut communiquer avec son processus enfant ✓ Oui (nous verrons cela plus tard dans le cours), mais il y a déjà une forme de communication car le fils récupère toutes les symboles du père

Degré de certitude ? : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

Les réponses correctes sont : Le processus parent connaît le processus enfant qu'il a engendré, Le processus parent peut communiquer avec son processus enfant, Un processus peut connaître son processus parent

Question 10


Correct

Évaluation du degré de certitude 3,00

Pondération 1,00

Quelle est la différence entre processus et threads (processus légers) ?

Veuillez choisir une réponse.

- ☐ 1. La pile d'exécution est commune entre les threads
- ☐ 2. Aucune
- ☒ 3. Les données du processus sont communes entre les threads  Oui, il y a partage des données entre les threads (ce qui n'est pas sans poser de problème quand on écrit).

Degré de certitude  : ☐ C = 1 (peu sûr : < 67%) ☐ C = 2 (moyennement sûr : > 67%) ☒ C = 3 (tout à fait sûr : > 80%)

Votre réponse est correcte.

La réponse correcte est : Les données du processus sont communes entre les threads