

Semestre : 1 ☒ 2 ☐

Session : Principale ☐

Rattrapage ☒

Unité d'enseignement : Systèmes et sécurité

Module (s) : Programmation systèmes et réseaux, *Virtualization & Sécurité*

Classe(s) : 4NIDS1

Nombre des question: 32

Nombre de pages : 6

Date : 21/07/2020 Heure : 9h45

Durée : 1heure

-----SEA-----

Q17. Une méthode pour assurer l'exclusion mutuelle est d'utiliser une variable de verrouillage partagée verrou or cette méthode n'assure pas :

- A. Famine
- B. Exclusion mutuelle
- C. Progression
- D. Interblocage

Q18. Lorsqu'un thread se termine en invoquant « exit »:

- A. Le système force la terminaison de son thread principal uniquement.
- B. Son état de terminaison est enregistré dans son bloc de contrôle de processus.
- C. Son père est adopté par le processus init.
- D. Le système force la terminaison de tous ses processus fils.

Q19. Au niveau des codes fils, les instructions qui viennent après l'instruction execl ou execlp :

- A. S'exécutent après execl ou execlp
- B. Ne seront jamais exécutées
- C. Exécutées que par le père
- D. Exécutées par le père et les fils

Q20. Considérez le bout de code suivant :

« int i=0 ; if(fork()) {i=i+1;} fork() ; i=i+1; printf("i=%d \n", i); ». Les valeurs de i affichées par le processus principal PP et le premier fils F1 de PP sont :

- A. PP : i = 2 et F1 : i = 2
- B. PP : i = 2 et F1 : i = 1
- C. PP : i = 1 et F1 : i = 2
- D. PP : i = 1 et F1 : i = 1

Q21. Pour attendre la fin d'un thread on utilise la fonction :

- A- pthread_self()
- B- pthread_join()
- C- pthread_id()
- D- pthread_pid()

Q22. Préciser le nombre de processus créés par le programme ci-dessous :
`intmain() {
 intcpt=0; while (cpt< 3) { if (fork() > 0) cpt++; else cpt=3; }
}`

- A- 3
- B- 4
- C- 5**
- D- 2

Q23. Quels est l'avantage du choix d'un petit quantum de temps pour l'algorithme d'ordonnancement Round Robin :

- A- Partage du segment de données.
- B- Partage du processeur.**
- C- Partage de la pile d'exécution.
- D- Partage de la mémoire.

Q24. Le PCB d'un processus contient :

- A- Le pointeur d'instructions, les registres mémoires, thread de contrôle, opération E/S , le vecteur d'état (PSW) et la priorité du processus.
- B- Le pointeur d'instructions, les registres mémoires, les fichiers ouverts, le PID, le PPID et la priorité du processus.
- C- Le pointeur d'instructions, les registres mémoires, l'état, opération E/S, le vecteur d'état (PSW) et la priorité du processus.**
- D- Le pointeur d'instructions, les registres mémoires, thread de contrôle, opération E/S, le vecteur d'état (PSW) et la priorité du processus.

Q25. Un processus a commuté depuis l'état actif vers l'état prêt. L'ordonnancement (ou le scheduling) doit être :

- A. Non-préemptif.**
- B. Préemptif.
- C. Le plus court travail (job) en premier (SJF).
- D. Round Robin (tourniquet).

Q26. Juste après un fork, quel élément suivant peut différer entre le processus père et le processus fils?

- A. les fichiers ouverts**
- B. le programme exécuté
- C. le PID
- D. les variables

Q27. Le descripteur de fichier standard "stdin":

- A. Peut être ouvert en lecture**
- B. Peut être ouvert en écriture
- C. Peut être ouvert en lecture et en écriture
- D. N'est pas accessible

Q28. Une section critique est:

- A. Un processus qui utilise une variable partagée
- B. Un thread qui utilise une variable partagée
- C. Une méthode qui permet à un seul processus à utiliser une variable partagée

D. Une suite d'instructions qui accèdent à des objets partagés

Q29. Laquelle de ces affirmations est vraie pour un tube (pipe) ?

A. Il assure la synchronisation entre les processus

B. Il permet une communication bidirectionnelle entre des processus dans la même machine

C. Il permet de faire communiquer des processus de machines différentes

D. Il utilise le mode synchrone pour la communication entre les processus

Q30 Dans le problème des producteur consommateur :

A. Deux consommateurs peuvent travailler en même temps

B. Deux producteurs peuvent travailler en même temps

C. Un producteur et un consommateur peuvent travailler en même temps

D. Un seul processus peut travailler à un moment donné

.Q31. "Une unité d'exécution rattachée à un processus, chargée d'exécuter une partie du processus" est :

A. Processus

B. Exclusion mutuelle

C. Thread

D. Mémoire partagée

Q32. Préciser le nombre de processus créer par le programme ci-dessous :

```
int main() { if(fork() || fork()) ;fork();}
```

A. 4

B. 5

C. 6

D. 7