1. Quel est l'utilité du PID en Linux?
2. Expliquer les concepts de processus de premier plan et d'arrière-plan en Linux.
3. Quel en est l'intérêt?
4. Quel est le rôle de la fonction "exec" en Linux?
5. Deux processus en parallèles peuvent-ils partager un fichier s'il est utilisé en lecture seulement.
6. Nommer le type de processus nécessitant une synchronisation.
7. Quel est le rôle d'un sémaphore?
8. Expliquer le mécanisme mis en oeuvre par les primitives du système, lors de la demande d'attribution et de libération d'une ressource utilisant un sémaphore.
9. Expliquer pourquoi le programme doit être non interruptible pendant l'exécution des primitives "wait" et "signal" (respectivement P(x) et V(x)).
10. Que représente la section critique d'un processus.
11. Expliquer comment un sémaphore peut permettre la gestion efficace de l'exclusion mutuelle.
12. Donner un exemple type de programme nécessitant la mise en place de mécanismes gérant l'exclusion mutuelle.
13. Expliquer comment un sémaphore peut-il résoudre la synchronisation de processus coopérants.
14. Décrire un mécanisme utilisé en Linux afin de permettre à un processus de transmettre un message à un autre processus.
15. Décrire le concept d'interblocage (dead lock).
16. Indiquer comment le S.E. peut-il corriger et prévenir cette situation.
17. Décrire les deux types de processus sous Linux.
18. Une commande est-elle un processus sous Linux.
19. Quel est le rôle du processus "swapper" en Linux ?
20. Qu'est-ce qui caractérise le mode multiprogrammation sans interruption ?
21. Quel en est l'inconvénient majeur?
22. Décrire le concept de préemption d'allocation.
23. Expliquer la mise en oeuvre de la multiprogrammation en temps partagé, notamment en ce qui concerne les interruptions.
24. Expliquer pourquoi le superviseur doit être un processus affecté à la plus haute priorité.
25. Quelles sont les conditions requises afin que le superviseur active un nouveau processus?