

ENSEEIH – SN - Apprentissage 1A

Examen Systèmes centralisés I

1H45 – Documents autorisés

1- Ecrire un script shell, nommé cpici.sh, qui copie tous les fichiers fournis en arguments dans le répertoire courant. (1)

`./cpici.sh REP1/fic11 REP1/fic12 REP2/fic21` copie les 3 fichiers indiqués dans le répertoire courant

2- Dans la commande « find », l'option `-perm -u=x` permet de sélectionner les fichiers exécutable (attribut x) pour le propriétaire.

En utilisant « find », écrire la ligne de commande qui permet de copier tous les fichiers exécutables de l'arborescence d'un répertoire donné, dans le répertoire courant. (2)

3- Ecrire un script shell, qui n'utilise pas la commande « find », et qui permet de copier tous les fichiers exécutables de l'arborescence du répertoire fourni comme 1^{er} argument, dans le répertoire fourni comme 2^{ème} argument (avec contrôle strict des arguments). (4)

4- On suppose disposer d'une fonction qui renvoie une valeur aléatoire comprise dans un intervalle [min ... max] : `int random(int min, int max)`.

Ecrire le programme qui génère 3 processus : un père, un fils et un petit-fils :

- Le petit-fils affiche son pid, s'endort durant N secondes (N : valeur aléatoire comprise entre 5 et 100), puis s'arrête.
- Le fils affiche son pid, s'endort durant M secondes (M : valeur aléatoire comprise entre 5 et 100), puis s'arrête.
- Le père s'endort durant 200 secondes, puis s'arrête. (4)

5- Dans quel état se trouve le petit-fils si son père se termine avant lui ($N > M$) ? Et dans quel état se trouve le fils lorsqu'il se termine (idem si petit-fils se termine avant le fils) ? (1)

Modifier le programme pour que ce second cas ne se présente plus.

Solution bloquante qui modifie le comportement des processus concernés (1)

Solution non bloquante qui ne modifie pas le comportement des processus concernés (2)

Il ne faut pas réécrire le programme complet. Il suffit de donner les nouvelles séquences en indiquant leur emplacement.

6-On dispose d'un exécutable, appelé « wait », qui affiche, toutes les secondes, le message « je suis toujours vivant » sur la sortie standard, et s'arrête au bout de N secondes (N est une valeur fournie en argument).

Réécrire le programme précédent pour que le petit-fils lance l'exécutable « wait » avec une valeur aléatoire comprise entre 5 et 100, tout en redirigeant les messages vers son père. Ce dernier lit les messages reçus et les affiche à l'écran, et s'arrête quand il n'y en a plus.

Que se passe-t-il au niveau du fils lorsque le petit-fils s'arrête ? Pourquoi ? (4)

7- La primitive alarm(int n) permet d'initialiser un « timer » avec la valeur n. Ce timer va décrémenter toutes les secondes, et va activer le signal SIGALRM lorsque le timer atteint la valeur 0 (idem qu'une minuterie qui sonne après une durée écoulée).

Compléter le code du fils de manière à ce qu'il s'arrête au bout de M secondes même s'il reste encore des messages à recevoir (M valeur aléatoire comprise entre 5 et 100).

Que se passe-t-il si le fils se termine avant le petit-fils ? (2)

8- On souhaite que le père envoie des messages à son fils à un rythme totalement différent du petit-fils.

A- Peut-il le faire en utilisant le même canal que le petit-fils ? Expliquer. (1)

B- Le père utilise un canal différent de celui du petit-fils. Comment faire pour que le fils puisse prendre connaissance à temps des messages émis par son père et par son fils ? Ecrire le code correspondant (celui du fils seulement). (2)