

Commencé le mercredi 28 février 2024, 15:16

État Terminé

Terminé le mercredi 28 février 2024, 15:19

Temps mis 2 min 50 s

Note 10,00 sur 10,00 (100%)

Question 1

Correct

Note de 1,00 sur 1,00

Quels sont les énoncés vrais pour les tubes de communication ?

Veuillez choisir une réponse.

- ☒ a. Les tubes de communication sont des fichiers spéciaux qui servent à faire communiquer des processus d'un même système d'exploitation. ✓
- ☐ b. Les données insérées dans un tube de communication sont organisées en messages de même taille.
- ☐ c. Une fin de fichier dans un tube de communication est atteinte lorsque le tube est vide.

Votre réponse est correcte.

La réponse correcte est :

Les tubes de communication sont des fichiers spéciaux qui servent à faire communiquer des processus d'un même système d'exploitation.

Question 2

Correct

Note de 1,00 sur 1,00

Considérez la commande « `ls | sort > data` ». La sortie erreur STDERR du processus exécutant « `ls` » est :

Veuillez choisir une réponse.

- ☐ a. le fichier data.
- ☐ b. le pipe.
- ☒ c. l'écran. ✓
- ☐ d. le clavier.

Votre réponse est correcte.

La réponse correcte est :

l'écran.

Question 3

Correct

Note de 1,00 sur 1,00

Si un processus exécute cette instruction « `dup2(fd=open("fich",O_WRONLY),1);` » alors

Veuillez choisir une réponse.

- ☒ a. la sortie standard du processus devient le fichier fich. ✓
- ☐ b. l'écriture en utilisant fd est redirigée vers l'écran.
- ☐ c. l'écriture dans fich ne peut être réalisée via 1.

Votre réponse est correcte.

La réponse correcte est :

la sortie standard du processus devient le fichier fich.

Question 4

Correct

Note de 1,00 sur 1,00

Considérez le code suivant : (<https://onlinegdb.com/B1kz-RzeO>)

```
int main() {
    int fd[2]; pipe(fd);
    if (fork() == 0) {
        dup2(fd[1],1);
        close(fd[1]); close(fd[0]);
        execlp("ls", "ls", NULL);
    }
    close(fd[1]); char c;
    while(read(fd[0],&c,1)>0) write(1,&c,1);
    close(fd[0]);
    wait(NULL);
    _exit(0);
}
```

Sélectionnez les énoncés corrects.

Veuillez choisir au moins une réponse.

- ☒ a. Juste avant l'appel à fork, le tube de communication a un seul écrivain et un seul lecteur. ✓
- ☒ b. Juste après l'appel à fork, le tube de communication a deux écrivains et deux lecteurs. ✓
- ☐ c. L'appel `dup2(fd[1],1)` n'affecte pas le nombre d'écrivains associés au tube de communication.

Votre réponse est correcte.

Les réponses correctes sont :

Juste avant l'appel à fork, le tube de communication a un seul écrivain et un seul lecteur.,

Juste après l'appel à fork, le tube de communication a deux écrivains et deux lecteurs.

Question 5

Correct

Note de 1,00 sur 1,00

Un processus crée, dans l'ordre, un tube anonyme (*int fd[2]; pipe(fd);*) et 3 threads. Il attend ensuite la fin des 3 threads avant de se terminer. Les threads ne ferment pas et ne dupliquent pas les descripteurs du tube.

Le nombre d'écrivains (descripteurs en écriture) dans le tube, juste après la création des 3 threads, est :

Veuillez choisir une réponse :

Veuillez choisir une réponse.

- ☐ a. 4
- ☐ b. 3
- ☐ c. 2
- ☐ d. aucune réponse
- ☒ e. 1 ✓

Votre réponse est correcte.

La réponse correcte est :

1

Question 6

Correct

Note de 1,00 sur 1,00

Supposez qu'un thread d'un processus exécute cette instruction « `dup2(fd=open("fich",O_WRONLY),1);` ». Quels sont les énoncés corrects qui correspondent à l'effet de l'exécution de cette instruction ?

Veuillez choisir une réponse :

Veuillez choisir une réponse.

- ☐ a. Le thread principal du processus aura toujours accès à l'écran via stdout (le descripteur 1).
- ☒ b. Les threads du processus n'ont plus accès à l'écran via stdout (le descripteur 1). ✓
- ☐ c. Seul le thread ayant exécuté l'instruction va perdre l'accès à l'écran via stdout (le descripteur 1).

Votre réponse est correcte.

La réponse correcte est :

Les threads du processus n'ont plus accès à l'écran via stdout (le descripteur 1).

Question 7

Correct

Note de 1,00 sur 1,00

La fonction kill permet exclusivement :

Veuillez choisir une réponse.

- ☐ a. d'envoyer un signal SIGKILL à un processus.
- ☒ b. d'envoyer un signal à un ou plusieurs processus. ✓
- ☐ c. de forcer la terminaison d'un processus fils.
- ☐ d. de suspendre l'exécution d'un processus fils.

Votre réponse est correcte.

La réponse correcte est :

d'envoyer un signal à un ou plusieurs processus.

Question 8

Correct

Note de 1,00 sur 1,00

Le masque des signaux d'un processus indique quels signaux, à destination du processus, à :

Veuillez choisir une réponse.

- ☐ a. ignorer
- ☐ b. capter.
- ☐ c. traiter en priorité en appliquant le traitement par défaut.
- ☒ d. conserver pour les traiter ultérieurement. ✓

Votre réponse est correcte.

La réponse correcte est :

conserver pour les traiter ultérieurement.

Question 9

Correct

Note de 1,00 sur 1,00

Lorsqu'un processus tente d'écrire dans un tube rompu (fermé en lecture), le signal SIGPIPE est envoyé au processus. Le traitement par défaut associé à ce signal est la terminaison du processus.

Peut-on éviter cette terminaison forcée ?

Oui, si

Veuillez choisir une réponse.

- ☐ a. le processus associe le traitement par défaut au signal SIGPIPE (signal(SIGPIPE, SIG_DFL)), avant le premier accès en écriture au tube.
- ☒ b. le processus associe le traitement SIG_IGN au signal SIGPIPE (signal(SIGPIPE, SIG_IGN)), avant le premier accès en écriture au tube. ✓

Votre réponse est correcte.

La réponse correcte est :

le processus associe le traitement SIG_IGN au signal SIGPIPE (signal(SIGPIPE, SIG_IGN)), avant le premier accès en écriture au tube.

Question 10

Correct

Note de 1,00 sur 1,00

Considérez le programme suivant: https://onlinegdb.com/B1xKX-Lkg_

```
int Continuer = 1;
void action (int SIG){
    switch(SIG)
    {
        case SIGUSR2 : printf("SIGNAL: SIGUSR2 reçu\n"); Continuer=0; break;
        case SIGINT : printf("SIGNAL: SIGINT reçu \n"); break;
        case SIGUSR1 : printf("SIGNAL: SIGUSR1 reçu \n"); break;
        default : printf("SIGNAL %d: Signal Inconnu\n", SIG);
    }
}
int main () {
    signal(SIGUSR2, action);
    signal(SIGINT, action);
    while (Continuer) {
        kill(getpid(), SIGINT);
        kill(getpid(), SIGUSR1);
        kill(getpid(), SIGUSR2);
    }
    _exit(0);
}
```

Comment ce processus va-t-il se terminer ?

Veuillez choisir une réponse.

- ☐ a. Le processus va se terminer par l'appel `_exit(0)`.
- ☐ b. La terminaison du processus va être forcée par le signal `SIGUSR2`.
- ☒ c. La terminaison du processus va être forcée par le signal `SIGUSR1`. ✓
- ☐ d. La terminaison du processus va être forcée par le signal `SIGINT`.

Votre réponse est correcte.

Le processus exécutant ce programmes réagit aux signaux `SIGUSR2` et `SIGINT` en exécutant la fonction `action`. Par contre, pour le signal `SIGUSR1`, il réagit en réalisant le traitement par défaut associé au signal `SIGUSR1` (c-à-d forcer la terminaison du processus).

La réponse correcte est :

La terminaison du processus va être forcée par le signal `SIGUSR1`.