

Semaine 4 – Questions sur les communications inter-processus (IPC)

1) Répondez par Vrai ou Faux en justifiant vos réponses.

a) Un processus est une entité produite après compilation

Réponse : Faux, car un processus est une image d'un programme en exécution

b) Un processus est une entité produite après le chargement d'un programme (binaire) en mémoire

Réponse : Vrai, car une fois terminé le chargement d'un programme en mémoire un processus est créé

2) Citez quatre événements qui provoquent l'interruption de l'exécution d'un processus en cours, dans le système UNIX:

a) la fin d'un quantum

b) une demande d'une E/S

c) l'arrivée d'un signal

d) la mise en attente par l'opération `sem_wait` d'un sémaphore

- 3) Répondez par Vrai ou Faux en justifiant vos réponses. À propos des descripteurs de fichier standard :
- a) le descripteur 0 peut être ouvert en lecture
Vrai
 - b) le descripteur 0 peut être ouvert en écriture
Faux, car il y a des permissions sur le stdin qui empêche l'écriture (read only)
 - c) le descripteur 1 peut être ouvert en lecture
Faux, car il y a des permissions sur le stdout qui empêche la lecture (write only)
 - d) le descripteur 1 peut être ouvert en écriture
Vrai
- 4) Laquelle ou lesquelles de ces affirmations sont vraies pour **un** tube (*pipe*, en anglais) ? Justifiez chacun de vos choix.
- a) Il permet une communication bidirectionnelle
Faux, si on n'utilise qu'un seul tube. Cependant, c'est vrai si on utilise **deux** tubes (p. 24).
 - b) Il permet de faire communiquer des processus de machines différentes
Faux, pour les tubes anonymes et nommés (p. 25). Pour faire communiquer des processus de machines différentes on doit utiliser par exemple les *sockets*.
 - c) Il peut être matérialisé dans le système de fichiers
Vrai, pour les tubes nommés (p. 25). Faux, pour les tubes anonymes.
 - d) Il peut être créé directement en ligne de commande entre deux processus
Vrai, pour les tubes anonymes (p. 17). Faux, pour les tubes nommés.

- 5) Laquelle de ces affirmations décrit le mieux la relation entre un signal et une interruption ?
- a) Une interruption est une sorte de signal
 - b) Tous les signaux sont la manifestation logicielle d'une interruption matérielle
 - c) Un signal est une sorte d'interruption
 - d) Certains signaux sont la manifestation logicielle d'une interruption matérielle

Réponse : c) un signal est un logiciel d'interruption du flux normal de l'exécution d'un processus (interruption logicielle)

- 6) On désire compter le nombre de mots uniques dans une liste. Le code suivant permet de faire cette tâche :

```
sort -u <f1 >f2 // Trie l'entrée standard en laissant une occurrence
                  // unique de chaque ligne

wc -l <f2        // word count : affiche sur la sortie standard le nombre
                  // de lignes dans l'entrée standard
```

Peut-on y arriver sans créer de fichier intermédiaire? Écrivez le code correspondant.

Réponse: `sort -u <f1 | wc -l`

7) Que fait le bout de code suivant:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
    int fd[2], i = 2;
    char ch[100];
    while (i)
    {
        pipe(fd);
        if (fork())
        {
            close(fd[0]);
            dup2(fd[1], 1);
            close(fd[1]);
            break;
        }
        else
        {
            close(fd[1]);
            dup2(fd[0], 0);
            close(fd[0]);
        }
        i--;
    }
    scanf("%s", ch);
    printf("%s\n", ch);
    exit(0);
}
```

Réponse :

pour $i=2$, le père crée un tube puis un fils. Il dirige sa sortie standard vers le tube puis il se met en attente de données du clavier pour les déposer sur le tube. Ensuite, il se termine.

Le fils dirige son entrée standard vers le tube puis crée un autre tube et son propre fils ($i=1$). Il dirige sa sortie standard vers le deuxième tube créé puis se met en attente de lecture de données du premier tube. Les données lues sont déposées sur le deuxième tube. Ensuite, il se termine.

Le petit fils dirige son entrée standard vers le tube. Il se met en attente de lecture de données du second tube. Il sort de la boucle car i devient nul. Les données lues sont affichées à l'écran. Enfin, il se termine.