

PROGRAMMATION CONCURRENTE

CPE Lyon – 3ETI

TRAVAUX PRATIQUES SEANCES 2 ET 3

`os.fork()` – `os.exec()` – `os._exit()` – `sys.exit()` – `os.wait()`

Exercice 1 - Introduction au `fork()` et `exec()`

Ecrire un programme permettant d'exécuter deux processus, chacun réalisant son propre traitement. Tester la fonction `os.execlp()` en écrivant un programme qui lance un autre programme.

Exercice 2 - L'appel à `fork()` dans une boucle

Ecrire un programme qui fait appel à la fonction `os.fork()` dans une boucle `for i in range(3)`.

A chaque itération le programme affichera les informations suivantes :

(*i* : *valeur_de_i*) je suis le processus : *pid*, mon pere est : *ppid*, retour : *retour*

où

- *valeur_de_i* est la valeur de l'indice *i* de la boucle
- *pid* est le PID du processus courant,
- *ppid* est le PID du père du processus courant,
- *retour* est la valeur du code retour de l'appel à la fonction `os.fork()`.

Dessiner l'arbre des processus correspondant à l'exécution de ce programme.

Exercice 3 – `os.fork()` & `os.exec()`

Ecrire un programme équivalent aux commandes *shell* suivantes :

- `who & ps & ls -l` Les commandes séparées par `&` s'exécutent *simultanément* [en parallèle].
- `who ; ps ; ls -l` Les commandes séparées par `;` s'exécutent *successivement* [séquentiellement]

Exercice 4 – Questions cours 4

Soit un système qui exécute le programme suivant :

```
import os
n=0
for i in range(1,5) :
    fils_pid = os.fork()    #1
    if (fils_pid > 0) :    #2
        os.wait()         #3
        n = i*2
        break;
print("n = ", n)          #4
sys.exit(0)
```

- Après la ligne étiquetée #2, dans le bloc d'exécution du `if`, on se retrouve dans quel processus, le père ou le fils ? Pour qui la valeur de `fils_pid` est-elle nulle ?
- Ce programme est-il déterministe? (Justifiez)
- Même question si l'on supprime la ligne étiquetée #3, justifiez.
- Si le programme est déterministe tel quel, indiquez exactement ce qui sera affiché à l'écran lors de son exécution. S'il n'est pas déterministe, donnez un des affichages possibles.
- L'appel à `os.fork()`, ligne étiquetée /*1*/, peut-il échouer? Pourquoi?

PROGRAMMATION CONCURRENTE

CPE Lyon – 3ETI

Exercice 5

Ecrire un programme dont le fonctionnement est le suivant :

- ✓ Il lit sur la ligne de commande (utiliser `sys.argv`) le nombre `N` de processus à créer.
- ✓ Il crée ces `N` processus en faisant `N` appels à `os.fork()`.
- ✓ Il se met en attente (appel à `pid_fils, etat = os.wait()`) de ces `N` processus fils et visualise leur identité (`pid_fils` et la valeur de `etat`) au fur et à mesure de leurs terminaisons.

Chacun des processus fils P_i réalise le traitement suivant :

- Il visualise son *pid* (`os.getpid()`) et celui de son père (`os.getppid()`),
- Il se met en attente pendant $2*i$ secondes (`time.sleep(2*i)`) et visualise la fin de l'attente,
- Il se termine par `sys.exit(i)`.

Exercice 6

Considérons le programme Python suivant :

```
1 import os, time, random, sys
2 for i in range(4) :
3     if os.fork() != 0 :
4         break
5 random.seed()
6 delai = random.randint(0, 4)
7 time.sleep(delai)
8 print("Mon nom est " + chr(ord('A')+i) + " j ai dormi " +
9       str(delai) + " secondes")
9 sys.exit(0)
```

1. Donnez l'arbre généalogique des processus engendrés par ce programme.
2. Quels sont les affichages possibles à l'écran.
3. Sans modifier les lignes de 2 à 7, modifiez ce programme de façon à ce que les processus fassent leur affichage par ordre alphabétique inversé du nom.

Exercice 7

Ajoutez à l'endroit indiqué dans le programme ci-dessous des instructions permettant de gérer les processus nécessaires pour provoquer l'affichage du message «**Bonjour**» exactement **4^N** fois. Vous ne pouvez pas insérer du code supplémentaire ailleurs qu'à l'endroit indiqué et vous ne pouvez plus ajouter d'instructions `print()`. Vous devez donner une solution utilisant des `os.fork()`, avec éventuellement des `sys.wait()`.

PROGRAMMATION CONCURRENTE

CPE Lyon – 3ETI

Donnez toutes les justifications nécessaires sous forme de commentaires dans le script.

```
N = 3  
for i in range(N) :  
    #_____début des ajouts_____  
  
    # _____fin des ajouts_____
```

print("Bonjour")

sys.exit(0)