TD : Vie et mort d'un processus

1) ouvrir un terminal, lancer la commande top (pour arrêter apurer sur q)

Combien y-a-t il de processus ?
Combien y-a-t-il de processus dans l'état running ? sleeping, stopped ? Zombie ?

Un processus peut donc se trouver dans différents états :

- prêt (*ready*): le processus attend son tour pour prendre la main
- en exécution (*running*): le processus a accès au processeur pour exécuter ses instructions
- en attente (*sleeping*) : le processus attend qu'un événement se produise (saisie clavier, réception d'une donnée par le réseau ou le disque dur ...)
- arrêté (*stopped*) : le processus a fini son travail ou a reçu un signal de terminaison (SIGTERM, SIGKILL, ...). Il libère les ressources qu'il occupe.
- zombie : Une fois arrêté, le processus informe son parent afin que ce dernier l'élimine de la table des processus. Cet état est donc temporaire mais il peut durer si le parent meure avant de pouvoir effectuer cette tâche. Dans ce cas, le processus fils reste à l'état zombie...

2) En utilisant https://debian-facile.org/doc:systeme:top expliquer chacune des colonnes .

Un processus est un programme en cours d'exécution sur un ordinateur. Il est caractérisé par

- un ensemble d'instructions à exécuter - souvent stockées dans un fichier sur lequel on clique pour lancer un programme (par exemple *firefox.exe*)
- un espace mémoire dédié à ce processus pour lui permettre de travailler sur des données qui lui sont propres : si vous lancez deux instances de *firefox*, chacune travaillera indépendamment de l'autre avec ses propres données.
- des ressources matérielles : processeur, entrées-sorties (accès à internet en utilisant la connexion Wifi).

3) ouvrir un autre terminal, taper xeyes. Quel est le PID de ce processus ?
Sur un autre terminal, taper renice +10 PID ou PID est le numéro de PID de xeyes.
Quel est l'effet de cette action ?

Un système de priorité permet à l'ordonnanceur de choisir le processus à élire.

L'ordonnanceur applique un algorithme prédéfini lors de la conception de l'OS. Le choix de cet algorithme va impacter directement la réactivité du système et les usages qui pourront en être fait. C'est un élément critique du système d'exploitation.

Sous Linux, on peut passer des consignes à l'ordonnanceur en fixant des priorités aux processus dont on est propriétaire : Cette priorité est un nombre entre -20 (plus prioritaire) et +20 (moins prioritaire).

4) ouvrir un autre terminal, tuer le processus xeyes (commande : kill 1756 si le PID est 1756)

Un processus est caractérisé par un identifiant unique : son **PID** (Process Identifier). Lorsqu'un processus engendre un fils, l'OS génère un nouveau numéro de processus pour le fils. Le fils connaît aussi le numéro de son père : le **PPID** (Parent Process Identifier).

5) dans un terminal lancer xeyes et dans un autre, taper la commande ps -eF
Trouver le PID et le PPID de xeyes.
Qui est le pére de xeyes ? Son grand-père ? Retrouver ainsi tous les ancêtre de xeyes.

Sur Linux, la création d'un processus se fait par clonage d'un autre processus au travers d'un appel systeme : **fork()**.

- le processus qui fait appel à **fork()** est appelé **processus père**
- le processus qui est ainsi créé par clonage est le **processus fils**

6) Dans un terminal, taper pstree puis retrouver xeyes dans l'arborescence.

Sur Linux, le père de tous les processus se nomme **init**, il est créé au démarrage du système.
Il se crée ensuite une arborescence de processus.

7) Dans un terminal, taper cat /proc/cpuinfo
Combien de processeur possède le Raspberry ?

8) Créer un programme python infiny.py avec une boucle infinie (utiliser les instructions touch pour créer le fichier et cat pour créer le programme)

Lancer ce programme sur plusieurs terminaux (taper python infiny.py)
A chaque fois observer la gestion des ressources de l'ordonnanceur . Compléter le tableau suivant :

Nombre de infny.py	%CPU
3	100 / 100 / 99.7
4	
5	
6	
7	

Peut-on prévoir les valeurs de la ligne suivante (8 processus infiny.py) ? pourquoi ?