**Linux – Fiche 13  
Processus**

1. Pour ce premier exercice, si vous le pouvez, travaillez localement sur votre machine Linux plutôt que sur le serveur *courslinux*.  
   Lancez la commande suivante **en arrière-plan** :

find / -type f -exec grep -Iq . {} \; -and -exec cat {} \;

find / -type f -exec grep -Iq . {} \; -and -exec cat {} \; &

* 1. Pendant que la commande s’exécute, listez les jobs.
  2. Y arrivez-vous ? Pour quelle raison ?
  3. Comment arrêter l’exécution de cette commande ? (sans fermer le terminal, ni en utiliser un autre !)

Fg

Ctrl c

* 1. Comment exécuter cette commande en arrière-plan sans que son affichage empêche   
     l’utilisation du terminal ?

find / -type f -exec grep -Iq . {} \; -and -exec cat {} \; > out.txt 2> err.txt &

*A la fin de cet exercice, n’oubliez pas d’effacer le(s) fichier(s) généré(s) au point d ! Il(s) peu(ven)t rapidement occuper plusieurs Gb d’espace disque.*

1. Lancez la commande sleep 600 en arrière-plan.

Sleep 600 &

* 1. Donnez la commande permettant de donner l’état de ce processus.

Jobs ou ps -u

* 1. Quel est cet état, et pourquoi ?
  2. Amenez-le à l’avant-plan.

Fg %1 (par defaut c’est %1 car y’en a que 1)

* 1. Suspendez-le (i.e. mettre en pause, pas terminer) et vérifiez son état.

Ctrl z

* 1. Relancez-le à l’arrière-plan.

bg

* 1. Listez les jobs.

jobs

* 1. Lancez d’autres commandes sleep en arrière-plan et listez ces jobs.

Sleep 500 &

Sleep 400 &

jobs

* 1. Ramenez le job n°2 à l’avant-plan et interrompez-le par CTRL+C. Affichez les jobs restants.

Fg %2 (%2 car il faut le n°2)

Ctrl c

jobs

* 1. Faites de même pour les jobs restants.

Fg 3

Fg (car il reste que le n°1)

1. Connectez-vous sur le serveur *courslinux* et téléchargez le programme timer :

- si vous êtes connectés au serveur *courslinux*, directement dans le répertoire de M. Legrand :   
 cp /home/anthony.legrand/timer .

- si vous travaillez localement, via la commande :   
 wget https://courslinux.ipl.be/~anthony.legrand/timer

Exécutez timer (en avant-plan) :

Pour exécuter timer :

* 1. Interrompez-le, est-ce que cela fonctionne ?

Ctrl c

* 1. Suspendez-le, est-ce que cela fonctionne ?

Ctrl

* 1. Que faire pour l’arrêter quand même ?

Se connecter sur un autre terminal

Ps -u pour voir qu’est-ce qui tourne

Kill -9 (le PID concerner)

1. Quel est le PID de votre bash ?

Ps -u et voir ou ps -l

Quel est son PPID ? Quel est ce processus ?

Ps -l et voir

1. En une seule ligne, affichez le PID et le chemin complet des processus en cours d’exécution et dont la commande contient la chaîne « systemd » (ou « init » si vous êtes sur Ubuntu WSL).

Allez-y par étape :

* 1. Affichez tous les processus s’exécutant sur le système.

Ps -ef ou ps - aux

* 1. Ne gardez que les lignes possédant « systemd » (ou « init »).

Ps- ef | grep systemd

* 1. Jetez la ligne possédant « grep » lui-même.

Ps- ef | grep systemd | grep -v grep

* 1. Quand il y a plusieurs espaces, n’en gardez qu’un seul.

Ps- ef | grep systemd | grep -v grep | tr s ‘ ‘

* 1. L’espace est donc un séparateur de champ, et il ne faut garder que les deux champs correspondant au PID et au chemin absolu du processus.

ps -ef | grep systemd | grep -v grep | tr s ' ' | cut -d ' ' -f2,8