



Linux et Bash - Processus

🕒 45 minutes 📺 Normal



DataScientest • com

Introduction à Linux

Machine status



Ubuntu
Server
18.04 LTS
SSD
Volume
Type
64-bit x86

Online

34.245.135.23



Connect

Reset



Stop

Processus

Qu'est ce qu'un processus

Une instance d'un programme est appelée un processus. En termes simples, toute commande que vous donnez à votre machine Linux démarre un nouveau processus. Une application sur votre système, un script Bash, ou encore une simple commande **ls** crée un nouveau processus. Plusieurs processus peuvent s'exécuter en même temps. Il existe deux types de processus :

- *Foreground processus* : Ils s'exécutent à l'écran et nécessitent une intervention de l'utilisateur (Office, votre navigateur web, application de retouche photos/videos...).
- *Background processus* : Ils s'exécutent en arrière-plan et n'ont généralement pas besoin de l'intervention de l'utilisateur (Antivirus, programme Python de web scraping...).

Exécution de processus

Foreground

Pour lancer un processus d'avant-plan, vous pouvez soit l'exécuter à partir du tableau de bord, soit l'exécuter à partir du terminal.

Si vous utilisez le terminal, vous devrez attendre que le processus de premier plan s'exécute.

Background

Si vous démarrez un programme/processus en avant-plan à partir du terminal, vous ne pouvez pas travailler sur le terminal tant que le programme n'est pas opérationnel.

Les tâches particulières, à forte intensité de données, nécessitent beaucoup de puissance de traitement et peuvent même prendre des heures. Vous ne voulez pas que votre terminal soit bloqué pendant un tel laps de temps.



Mise en situation

Imaginons que le script Python suivant permet de faire du webscraping. Chaque seconde, il récupère des données et les écrit dans le fichier `data.txt`.

```
1 import time
2 i = 0
3 while i < (60 * 20):
4     fichier = open("data.txt", "a")
5     fichier.write(str(i) + "\n")
6     fichier.close()
7     i += 1
8     time.sleep(1)
9
```

Recopiez ce script dans un fichier `webscraping.py` et exécutez.

```
1 python3 webscraping.py
2
```

Vous venez de lancer votre script en avant-plan. Le problème c'est que le script va tourner pendant 20 minutes. Nous aimerions bien avoir un certain contrôle sur notre terminal pendant ce temps. Pour cela, il existe plusieurs commandes pour interagir avec notre processus.

À chaque commande que nous allons voir, prenez le temps d'observer les dernières lignes du fichier `data.txt` (par exemple `tail -3 data.txt`).

Mise en pause

Avant tout pour récupérer le contrôle sur notre terminal il faut mettre notre processus en pause. Pour mettre un processus qui tourne en avant-plan en pause il suffit d'appuyer sur les touches de votre clavier `Ctrl+z`.

```
1 [1]+  Stopped                  python3 webscraping.py
2
```

La ligne ci-dessus devrait s'afficher (observez le contenu de `data.txt`).

Mise en Foreground

Nous pourrions vouloir mettre un processus en avant-plan pour cela vous pouvez utiliser la commande `fg`. Exécutez la commande pour voir que celle-ci fonctionne bien puis remettez vous en pause.

Mise en Background

Vous vous en doutez certainement la commande pour mettre en arrière-plan c'est `bg`. Grâce à celle-ci le processus continue de tourner en laissant le terminal entre vos mains.

```
1 [1]+ python3 webscraping.py &
2
```

La ligne ci-dessus devrait s'afficher. Nous observons le caractère `&` à la suite de notre commande d'exécution. Pour généraliser, si nous souhaitons directement lancer un script en arrière-plan, il suffit d'ajouter `&` à la suite de notre commande.

Outils supplémentaires

Il existe différentes commandes pour avoir des informations sur les processus en cours :



Linux. Lancez la commande `htop`, énormément d'informations s'affichent devant vous. Et oui ! Beaucoup de processus tournent sans que vous le sachiez afin de faire fonctionner votre système. Explorez l'outil afin de retrouver notre processus qui doit certainement être en train de tourner via la commande `man`.

Appuyez sur `q` sur votre clavier pour sortir de l'affichage du processus `htop`.

PS

Cette commande signifie "Process Status" (État du processus). Cette commande est similaire à la commande `htop` mais les informations affichées sont différentes.

Pour vérifier tous les processus en cours d'exécution sous un utilisateur, utilisez la commande `ps ux`. On observe ainsi plus facilement les informations concernant les processus lancés par nous-même.

PID

Avec les outils précédents, une des informations concernant notre processus est le PID pour Process ID. Cet identifiant nous permet de pouvoir interagir directement avec le processus. Vous pouvez par exemple faire un `ps PID` pour avoir seulement les informations du processus concerné, ou bien vous pouvez décider de mettre fin au processus.

Pour trouver facilement le PID d'un processus vous pouvez utiliser la commande `pidof Process_Name`. Dans notre exemple : `pidof python3 webscraping.py`

Mettre fin au processus : Kill

La commande `kill` met fin aux processus en cours d'exécution sur une machine Linux. Pour l'utiliser, vous devez connaître le PID (identifiant du processus) du processus que vous voulez tuer et exécuter la commande `kill PID`.

Validated