

# Mettre un processus en arrière plan

(basé sur <https://linuxize.com/post/how-to-run-linux-commands-in-background/>)

## L'esperluette &

Il suffit de rajouter esperluette & à la fin de la commande pour garder la main sur son terminal :

```
nom_commande &
```

Une autre syntaxe fréquente est :

```
nom_commande > /dev/null 2>&1 &
```

qui renvoie la sortie standard **stdout** vers **/dev/null** (évite que les messages du processus ne viennent polluer le terminal en cours), et qui redirige également **stderr** (2) vers **stdout** (1), ce qui a pour effet de cacher également les messages d'erreur du processus...

## La liste des processus

Pour avoir la liste des processus mis en arrière plan :

```
jobs -l
```

## Rappeler un processus

```
fg $numéro_renvoyé_par_jobs
```

Si vous n'avez qu'un seul processus en arrière plan, la commande se réduit à fg.

## Arrêter temporairement un processus et le mettre en arrière-plan

Le raccourci clavier pour arrêter un processus bloquant le terminal est :

```
CTRL+Z
```

Le processus est alors placé en arrière plan et marqué comme **Arrêté** dans jobs.

Comme le terminal est de nouveau accessible, on peut ensuite choisir de mettre le programme réellement en arrière plan, ce qui a aussi pour effet de le remettre en route :

```
bg $numéro_renvoyé_par_jobs
```

## Détacher un programme de son terminal parent

---

Par défaut, les processus lancés avec `&` ou placés en arrière-plan sont attachés au terminal courant. Si on ferme le terminal, les programmes liés sont fermés avec.

Si on veut détacher un programme du terminal, il faut soit le lancer dès le départ avec `nohup` comme suit :

```
nohup nom_commande &
```

Le fait d'utiliser `nohup` permet ainsi d'éviter le signal `SIGHUP` (« hangup » ou « raccroché »).

Soit, si le processus est déjà lancé, utilisé `disown` :

```
disown $numéro_renvoyé_par_jobs
```

S'il n'y a qu'un processus en arrière plan, la commande se réduit en `disown`.

## Tuer un processus

---

Rappel : pour obtenir le **process id** ou **id de processus**, c'est `ps aux|grep nom_processus`, et pour la liste des processus en arborescence, c'est `ps tree`.

Outre le signal **SIGHUP** (1) déjà vu, on utilise généralement **SIGTERM** (15) et **SIGKILL**(9) pour tuer « gentiment » ou « violemment » un processus.

```
kill -15 $PID
```

ou

```
kill -9 $PID
```

Pour la liste de tous les signaux disponibles, c'est `kill -l`.

## Tuer plusieurs processus

---

Quand on a lancé plusieurs fois un même processus, il est plus simple d'utiliser :

```
killall $nom_processus
```

Là encore, l'option `-s` de la commande permet de fixer le signal à envoyer (`SIGTERM` par défaut).