

## Ctrl + C : arrêter un processus lancé en console

Parfois, rien ne va plus. Un processus s'emballé et ne veut pas s'arrêter. Cela arrive partout, même sous Linux. À la différence de Windows toutefois, vous ne devriez pas avoir le réflexe de redémarrer « pour que ça aille mieux ». Tout peut être résolu en arrêtant les processus qui vous gênent et en les relançant au besoin.

Il y a plusieurs façons d'arrêter un processus, nous allons les étudier ici.

La combinaison de touches `Ctrl + C` est à connaître. Cela demande (gentiment) l'arrêt du programme console en cours d'exécution à l'écran. Ce raccourci se comporte ainsi en mode console seulement. En effet, en mode graphique, le comportement est le même que sous Windows : cela permet d'effectuer une copie dans le presse-papier. Notez que pour copier-coller sous Linux, on utilise souvent une autre technique : on sélectionne du texte et on clique avec la molette de la souris pour le coller ailleurs.

Prenez une commande qui n'en finit plus, comme par exemple un `find` sur l'ensemble du disque. Celui-ci va analyser tout votre disque dur à la recherche du fichier demandé. Si vous trouvez cela trop long et que vous voulez arrêter le programme en cours de route, il vous suffit de taper `Ctrl + C` :

```
# find / -name "*log*"
/dev/log
/bin/login
/sys/module/scsi_mod/parameters/scsi_logging_level
/sys/module/ehci_hcd/parameters/log2_irq_thresh
```

La liste aurait dû être beaucoup plus longue. Mais j'ai demandé l'arrêt du programme avec `Ctrl + C`, ce qui fait que j'ai pu « retrouver » l'invite de commandes rapidement et facilement.

Taper `Ctrl + C` ne coupe pas le programme brutalement, cela lui demande gentiment de s'arrêter, comme si vous aviez cliqué sur la croix pour fermer une fenêtre.

## **kill** : tuer un processus

`Ctrl + C` ne fonctionne que sur un programme actuellement ouvert dans la console. De nombreux programmes tournent pourtant en arrière-plan, et `Ctrl + C` n'aura aucun effet sur eux.

C'est là que vous devez utiliser `kill` si vous voulez les arrêter (on dit aussi « tuer », c'est pareil même si ça a l'air violent).

Pour vous en servir, il faudra auparavant récupérer le PID du ou des processus que vous voulez tuer. Pour cela, deux solutions :

- `ps` ;
- `top`.

Ces deux commandes que nous venons de voir vous indiquent le PID (numéro d'identification) de chaque processus. Par exemple avec `ps` :

```
$ ps -u mateo21
```

PID	TTY	TIME	CMD
5012	?	00:00:01	gnome-session
5057	?	00:00:00	ssh-agent
5060	?	00:00:00	dbus-launch
5061	?	00:00:00	dbus-daemon
5063	?	00:00:03	gconfd-2
5066	?	00:00:00	gnome-keyring-d
5068	?	00:00:03	gnome-settings-
5075	?	00:00:00	sh
5076	?	00:00:00	esd
5080	?	00:00:26	metacity
5083	?	00:00:17	gnome-panel

...

25227	pts/1	00:00:00	bash
32617	pts/1	00:00:00	man
32627	pts/1	00:00:00	pager
32703	pts/0	00:00:00	ps

Supposons qu'on souhaite arrêter Firefox. On peut filtrer cette longue liste avec `grep` et un pipe que nous avons appris à utiliser.

```
$ ps -u mateo21 | grep firefox
32678 ?          00:00:03 firefox-bin
```

Hop là, on a filtré Firefox de cette longue liste et on a même récupéré son PID. Il ne nous reste plus qu'à le tuer, avec la commande suivante :

```
kill 32678
```

Si tout va bien, la commande ne renvoie rien. Sinon, une erreur devrait s'afficher dans la console.

Vous pouvez aussi tuer plusieurs processus d'un seul coup en indiquant plusieurs PID à la suite :

```
kill 32678 2768 33071
```

Attention : même si `kill` est par défaut une commande « gentille » qui demande simplement au processus de s'arrêter, évitez de tuer des processus que vous ne connaissez pas. Beaucoup d'entre eux sont essentiels au bon fonctionnement de votre système, surtout ceux qui ont été lancés par root.

*J'ai essayé, mais Firefox a l'air vraiment complètement planté et il refuse de s'arrêter. Il n'y a pas moyen d'être un peu plus... direct ?*

Vous voulez tuer un processus sans lui laisser le choix ?

C'est tout à fait possible, mais à n'utiliser que dans le cas d'un programme complètement planté que vous voulez vraiment arrêter !

Avec `kill -9` (comme le chiffre 9, oui, oui), vous demandez à Linux de tuer le processus sans lui laisser le temps de s'arrêter proprement. Cela peut faire le ménage quand rien ne va plus.

```
kill -9 32678
```

... tuera le processus n°32678 (Firefox, dans mon cas) immédiatement sans lui laisser le temps de finir.

## **killall : tuer plusieurs processus**

Souvenez-vous : je vous ai dit que certains programmes se dupliquaient en plusieurs processus. Si vous voulez arrêter l'ensemble de ces processus, comment faire ? Heureusement, vous avez des armes pour éradiquer cette vermine.

Vous pourriez, certes, tuer tous les processus en récupérant un à un leur PID. Mais il y a plus rapide : `killall` (« tuez-les tous ! »).

Contrairement à `kill`, `killall` attend le nom du processus à tuer et non son PID.

Supposons que nous ayons trois processus `find` en cours d'exécution que nous souhaitons arrêter.

```
$ ps -u mateo21 | grep find
 675 pts/1      00:00:01 find
 678 pts/2      00:00:00 find
 679 pts/3      00:00:01 find
```

Pour tous les tuer, il faudra donc taper :

```
$ killall find
```

Si la commande ne renvoie rien, c'est que tout s'est bien passé.

En revanche, si vous avez :

```
$ killall find
find: aucun processus tué
```

... cela signifie qu'il n'y avait aucun processus de ce nom à tuer. Soit le processus n'est plus là, soit vous n'avez pas écrit correctement son nom. Vérifiez ce nom à nouveau avec la commande `ps`.