```
$ echo $$
```

Le PID sous Unix

À quoi sert le numéro d'identification d'un processus ?

D'abord : qu'est-ce qu'un processus ?

`/usr/bin/nano`

Le programme

Un fichier exécutable, posé sur le disque. Il ne fait rien tant qu'on ne le lance pas.

`nano en cours d'exécution`

Le processus

Une instance du programme en mémoire, avec son propriétaire, ses fichiers ouverts, son état… et un **numéro**.

Lancer trois fois nano = un seul programme, mais **trois processus**. Il faut bien pouvoir les distinguer.

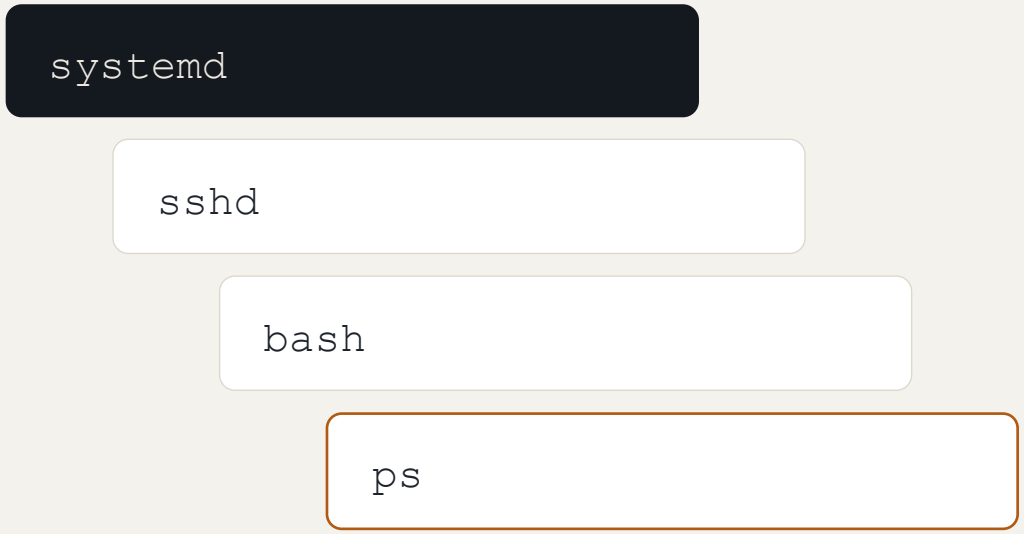
Le PID : la « plaque d'immatriculation » du processus

2041

Process IDentifier

- Un **entier positif** attribué par le noyau au lancement
- **Unique** parmi les processus vivants à un instant donné
- C'est par lui que le système et l'administrateur **désignent** un processus
- Libéré à la fin du processus, puis **réutilisable** plus tard

PID et PPID : une arborescence



Chaque processus est créé par un **parent**. Son **PPID** (Parent PID) est le PID de ce parent.

Le **PID 1** (systemd ou init) est l'ancêtre de tous les processus utilisateur.

```
$ pstree -p
```

Usage 1 : retrouver et observer un processus

Commande	Ce qu'elle donne
ps aux	Tous les processus avec leur PID, propriétaire, CPU, mémoire
pgrep -a sshd	Le ou les PID correspondant à un nom
top / htop	Vue en temps réel, triée par consommation
echo $$ · echo $!	PID du shell courant · PID du dernier processus lancé en arrière-plan
ls /proc/2041/	Tout ce que le noyau sait du processus 2041 : cmdline, environ, fd…

Usage 2 : envoyer un signal à un processus

La commande kill -SIGNAL PID cible un processus **par son PID**.

Commande	Signal	Effet
kill 2041	SIGTERM (15)	Demande poliment de s'arrêter (par défaut)
kill -9 2041	SIGKILL (9)	Arrêt forcé par le noyau, impossible à ignorer
kill -HUP 812	SIGHUP (1)	Souvent : recharger la configuration d'un service
kill -STOP / -CONT	SIGSTOP / SIGCONT	Mettre en pause, puis reprendre

La vie d'un PID

`fork()`

Naissance

Le parent se duplique : l'enfant reçoit un nouveau PID.

`exec()`

Vie

L'enfant charge un autre programme, en gardant son PID.

`exit()`

Zombie

Terminé, il attend que le parent lise son code retour (état Z).

`wait()`

Libération

Le PID est rendu au noyau et pourra être réattribué.

Les PID montent jusqu'à une limite (/proc/sys/kernel/pid_max), puis reprennent depuis le bas.

Pourquoi c'est utile en cybersécurité

Repérer l'intrus

Un processus inconnu ? Son PID mène à son binaire (/proc/PID/exe), son parent et son propriétaire.

Relier réseau et processus

ss -tulpn indique quel PID écoute sur quel port.

Fichiers PID des services

Les démons notent leur PID dans /run/*.pid pour être pilotés. Un fichier modifiable par tous est une faille.

Conteneurs

Grâce aux PID namespaces, un conteneur Docker a son propre PID 1, isolé de l'hôte.

À retenir

01 Le PID est le **numéro unique** qui identifie un processus vivant.

02 Il sert à **observer** (ps, /proc) et à **agir** (kill, renice).

03 Avec le PPID, il dessine l'**arbre** des processus, à partir du PID 1.

04 Il est **recyclé** : un PID ne désigne un processus que tant qu'il vit.

```
$ man ps · man kill · man proc
```